

**PENGEMBANGAN INTERNET *GATEWAY DEVICE* BERBASIS  
KONEKSI GPRS UNTUK MENGOLEKSI DAN MENERUSKAN  
DATA KE MEDIA PENYIMPANAN**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun oleh:

Maxi Luckies Ginanjar

NIM: 145150200111172



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018**

## PENGESAHAN

PENGEMBANGAN INTERNET GATEWAY DEVICE BERBASIS KONEKSI GPRS UNTUK  
MENGOLEKSI DAN MENERUSKAN DATA KE MEDIA PENYIMPANAN

### SKRIPSI

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Komputer

Disusun Oleh :  
Maxi Luckies Ginanjar  
NIM: 145150200111172

Skripsi ini telah diuji dan dinyatakan lulus pada  
1 Agustus 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Eko Sakti Pramukantoro, S.Kom, M.Kom  
NIK: 201102 860805 1 001

Dosen Pembimbing II



Reza Andria Siregar, S.T., M.Kom  
NIP: 19790621 200604 1 003

Mengetahui  
Ketua Jurusan Teknik Informatika



Tri Astoro Kurniawan, S.T, M.T, Ph.D  
NIP: 19710518 200312 1 001

## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 1 Agustus 2018



Maxi Luckies Ginanjar

NIM: 145150200111172

## KATA PENGANTAR

Assalamualaikum warahmatullahi wabarakatuh.

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga laporan skripsi yang berjudul “Pengembangan Internet *Gateway Device* Berbasis Koneksi GPRS Untuk Mengoleksi dan Meneruskan Data ke Media Penyimpanan” dapat terselesaikan.

Selama pengerjaan laporan skripsi penulis mendapat banyak pelajaran dan pengalaman baru. Penulis juga banyak mengaplikasikan ilmu yang didapat selama menduduki bangku perkuliahan ke dalam laporan skripsi ini. Penulis menyadari bahwa laporan skripsi ini tidak akan berhasil tanpa dukungan, doa, dan bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa erima kasih dan rasa hormat kepada :

1. Bapak Eko Sakti Pramukantoro, S.Kom.,M.Kom., dan bapak Reza Andria Siregar, S.T.,M.Kom., selaku dosen pembimbing skripsi penulis yang begitu sabar membimbing penulis juga memberikan arahan dengan sangat baik kepada penulis sehingga dapat menyelesaikan skripsi ini.
2. Kedua orang tua dan keluarga besar saya yang tidak henti-hentinya memberikan doa dan dukungan selama proses menempuh pendidikan di Malang termasuk saat proses pengerjaan skripsi ini.
3. Bapak Wayan Firdaus Mahmudy, S.Si.,M.T., Ph.D., selaku Dekan Fakultas Ilmu Komputer
4. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D., selaku Ketua Jurusan Teknik Informatika
5. Bapak Agus Wahyu Widodo, S.T., M.Cs., selaku Ketua Program Studi Teknik Informatika
6. Tuti Wardani Hamid, Binariyanto Aji, Hilman Nihri, Uis Yudha Tri Wirawan, Landika Hari Suganda, Ahmad Naufal, Jessy Ratna W, yang selalu meluangkan waktu untuk membantu dan berbagi ilmu sehingga laporan skripsi ini dapat terselesaikan
7. Seluruh anggota keluarga L-Tifam yang selalu memberikan bantuan, saran, dan semangat mulai awal perkuliahan di Malang sampai dengan proses pengerjaan skripsi ini selesai
8. Teman-teman anggota Onigiri dan Teletubier, yang telah memberikan banyak pengalaman, semangat, doa dan dukungan moral
9. Seluruh civitas akademika Fakultas Ilmu Komputer yang selalu memberikan dukungan dan bantuan dalam proses penyelesaian skripsi ini



10. Teman-teman keminatan KBJ dan teman – teman Teknik Informatika Angkatan 2014 lainnya yang selalu berbagi ilmu dari awal perkuliahan sampai tahap penyelesaian skripsi
11. Semua pihak yang telah banyak membantu, berbagi ilmu dan pengalaman, serta memberikan dukungan selama proses pengerjaan skripsi yang tidak dapat penulis sebutkan satu per satu

Penulis mengakui bahwa dalam penyusunan skripsi ini masih terdapat banyak kekurangan, sehingga kritik yang membangun dan juga saran sangat penulis butuhkan. Akhir kata penulis berharap agar skripsi ini bisa membawa manfaat bagi semua pihak yang menggunakannya.

Wassalamualaikum warahmatullahi wabarakatuh.

Malang, 1 Agustus 2018

Penulis

[luckiesmaxi6@gmail.com](mailto:luckiesmaxi6@gmail.com)



## ABSTRAK

**Maxi Luckies Ginanja, Pengembangan Internet *Gateway Device* Berbasis Koneksi GPRS untuk Mengoleksi dan Meneruskan Data ke Media Penyimpanan**

**Dosen Pembimbing: Eko Sakti P., S.Kom., M.Kom. dan Reza Andria Siregar, S.T., M.Kom.**

Penerapan lingkungan *IoT* untuk memonitor lingkungan perkantoran menemui masalah ketika data dari *middleware* tidak dapat diteruskan ke media penyimpanan karena masih terletak pada jaringan *intranet*. Masalah tersebut dapat diatasi dengan menambahkan *internet gateway device* agar sistem yang telah bekerja pada jaringan *intranet* dapat mengirimkan data ke jaringan *internet* yang lebih luas. Dengan menambahkan modul GSM pada *internet gateway device* penelitian sebelumnya, penelitian ini mengusulkan dua skenario penerimaan dan pengiriman data berbasis koneksi GPRS. Hasil dari penelitian ini menunjukkan bahwa *internet gateway device* dapat diterapkan menggunakan koneksi GPRS dengan rata-rata *throughput* untuk skenario pengiriman pertama sebesar 915,05 bps dan skenario kedua sebesar 347,75 bps.

Kata kunci: *gateway*, GPRS, GSM, *internet of things*

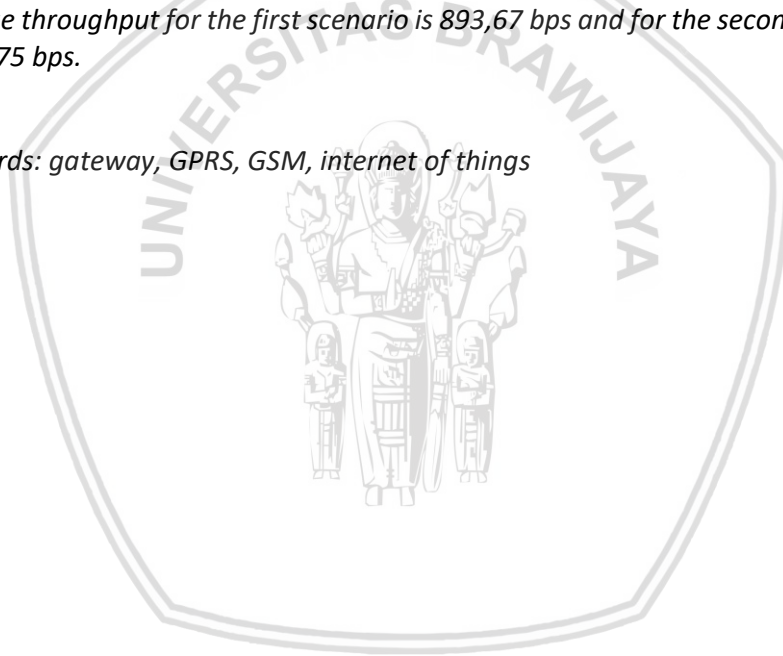
## ABSTRACT

**Maxi Luckies Ginanja, Development of Internet Gateway Device Based on GPRS Connection for Collecting and Forwarding data to The Storage Media**

**Supervisors : Eko Sakti P., S.Kom., M.Kom. dan Reza Andria Siregar, S.T., M.Kom.**

*Implementation of IoT system for monitoring office environment faces a problem when data from the middleware can't be forwarded to data storage located on the internet because it still located in the intranet network. This problem can be solved by adding an internet gateway device so that the previous system can communicate with a wider internet network. In this research, a GSM module added to the internet gateway device from previous work and also two scenarios are proposed for collecting and sending data with GPRS connection. The results show that internet gateway device can be implemented with GPRS connection with average throughput for the first scenario is 893,67 bps and for the second scenario is 347,75 bps.*

**Keywords:** gateway, GPRS, GSM, internet of things



## DAFTAR ISI

|  |      |
|--|------|
| PENGESAHAN .....                                       | ii   |
| PERNYATAAN ORISINALITAS .....                          | iii  |
| KATA PENGANTAR .....                                   | iv   |
| ABSTRAK .....  | vi   |
| ABSTRACT .....   | vii  |
| DAFTAR ISI .....                                       | viii |
| DAFTAR TABEL .....                                     | xi   |
| DAFTAR GAMBAR .....                                    | xii  |
| BAB 1 PENDAHULUAN .....                                | 1    |
| 1.1 Latar belakang .....                               | 1    |
| 1.2 Rumusan masalah .....                              | 3    |
| 1.3 Tujuan .....                                       | 3    |
| 1.4 Manfaat .....                                      | 3    |
| 1.5 Batasan masalah .....                              | 3    |
| 1.6 Sistematika pembahasan .....                       | 4    |
| BAB 2 LANDASAN KEPUSTAKAAN .....                       | 5    |
| 2.1 Kajian Pustaka .....                               | 5    |
| 2.2 Internet of Things .....                           | 6    |
| 2.3 Internet <i>Gateway Device</i> .....               | 6    |
| 2.4 Global System for Mobile communication (GSM) ..... | 6    |
| 2.5 General Packet Radio <i>Service</i> (GPRS) .....   | 7    |
| 2.6 JavaScript Object Notation (JSON) .....            | 7    |
| 2.7 <i>Throughput</i> .....                            | 7    |
| 2.8 <i>Delay</i> .....                                 | 7    |
| BAB 3 METODOLOGI .....                                 | 8    |
| 3.1 Studi Literatur .....                              | 9    |
| 3.2 Analisis Kebutuhan .....                           | 9    |
| 3.3 Perancangan .....                                  | 9    |
| 3.4 Implementasi .....                                 | 9    |
| 3.5 Pengujian dan Pembahasan Hasil Pengujian .....     | 10   |

|  |    |
|--|----|
| 3.6 Pengambilan Kesimpulan dan Saran.....                            | 10 |
| BAB 4 Analisis Kebutuhan .....                                       | 11 |
| 4.1 Deskripsi Umum Sistem.....                                       | 11 |
| 4.2 Batasan .....  | 11 |
| 4.3 Lingkungan Operasi.....  | 11 |
| 4.4 Kebutuhan Jaringan.....  | 11 |
| 4.5 Kebutuhan Data .....   | 12 |
| 4.6 Kebutuhan Sistem .....   | 12 |
| 4.6.1 Kebutuhan Fungsional .....                                     | 12 |
| 4.6.2 Kebutuhan Nonfungsional .....                                  | 13 |
| BAB 5 PERANCANGAN DAN IMPLEMENTASI .....                             | 14 |
| 5.1 Perancangan .....  | 14 |
| 5.1.1 Perancangan Alur Sistem .....                                  | 14 |
| 5.1.1.1 Alur skenario dengan rentang waktu subscribe.....            | 15 |
| 5.1.1.2 Alur skenario pengiriman langsung .....                      | 17 |
| 5.1.2 Perancangan Pengalamatan dan Arsitektur Jaringan.....          | 18 |
| 5.1.3 Perancangan Pengujian.....                                     | 18 |
| 5.1.3.1 Pengujian fungsional.....                                    | 18 |
| 5.1.3.2 Pengujian nonfungsional.....                                 | 20 |
| 5.2 Implementasi .....   | 21 |
| 5.2.1 Implementasi jaringan .....                                    | 21 |
| 5.2.2 Implementasi <i>subscriber</i> .....                           | 22 |
| 5.2.2.1 Skenario dengan rentang waktu subscribe.....                 | 22 |
| 5.2.2.2 Skenario pengiriman langsung .....                           | 23 |
| 5.2.3 Implementasi pengiriman data dengan HTTP POST.....             | 24 |
| 5.2.3.1 Skenario dengan rentang waktu subscribe.....                 | 25 |
| 5.2.3.2 Skenario pengiriman langsung .....                           | 25 |
| BAB 6 Pengujian dan Pembahasan hasil pengujian.....                  | 27 |
| 6.1 Pengujian Fungsional .....                                       | 27 |
| 6.1.1 Pengujian <i>subscribe</i> tanpa definisi topik .....          | 27 |
| 6.1.2 Pengujian <i>subscribe</i> topik yang telah didefinisikan..... | 28 |
| 6.1.3 Pengujian penerimaan data JSON .....                           | 29 |



|  |    |
|--|----|
| 6.1.4 Pengujian pengiriman data secara berkala .....     | 29 |
| 6.1.5 Pengujian pengiriman data secara langsung .....    | 30 |
| 6.1.6 Pengujian pengiriman data dengan HTTP header ..... | 31 |
| 6.2 Pengujian Nonfungsional.....                         | 32 |
| 6.2.1 Pengujian <i>delay</i> dan <i>throughput</i> ..... | 32 |
| 6.2.2 Pengujian <i>reliability</i> .....                 | 35 |
| 6.2.3 Pengujian <i>scalability</i> .....                 | 38 |
| BAB 7 KESIMPULAN DAN SARAN .....                         | 40 |
| 7.1 Kesimpulan .....                                     | 40 |
| 7.2 Saran.....   | 40 |
| DAFTAR PUSTAKA .....                                     | 42 |



## DAFTAR TABEL

|  |    |
|--|----|
| Tabel 4.1 <i>Payload</i> data sensor DHT11 dan DHT22 .....           | 12 |
| Tabel 4.2 Skema <i>payload</i> data sensor kamera.....               | 12 |
| Tabel 4.3 Kebutuhan Fungsional Sistem .....                          | 13 |
| Tabel 4.4 Kebutuhan Nonfungsional Sistem .....                       | 13 |
| Tabel 5.1 Skenario pengujian fungsional .....                        | 18 |
| Tabel 5.2 Deskripsi pengujian nonfungsional .....                    | 20 |
| Tabel 6.1 Hasil pengujian <i>subscribe</i> dengan rentang waktu..... | 33 |
| Tabel 6.2 Hasil pengujian pengiriman langsung.....                   | 34 |
| Tabel 6.3 Hasil pengujian <i>reliability</i> .....                   | 35 |
| Tabel 6.4 Hasil pengujian <i>scalability</i> .....                   | 38 |



## DAFTAR GAMBAR

|  |    |
|--|----|
| Gambar 2.1 Gambaran sistem penyimpanan berbasis MongoDB dan GridFS .....                           | 6  |
| Gambar 3.1 Tahap Pengerjaan Penelitian .....   | 8  |
| Gambar 5.1 Diagram alir sistem secara umum .....   | 15 |
| Gambar 5.2 Rancangan alur skenario pertama.....  | 16 |
| Gambar 5.3 Rancangan alur skenario kedua .....   | 17 |
| Gambar 5.4 Arsitektur jaringan yang digunakan pada penelitian ini.....                             | 18 |
| Gambar 5.5 Skema Koneksi GPIO dan modul sim 808 .....  | 21 |
| Gambar 5.6 Program untuk menyalakan modul SIM 808 .....  | 22 |
| Gambar 5.7 Konfigurasi cronjob pada perangkat Internet <i>Gateway</i> .....                        | 22 |
| Gambar 5.8 Program untuk menentukan <i>topic</i> .....   | 22 |
| Gambar 5.9 Program <i>subscribe</i> dengan skenario waktu tunggu .....                             | 23 |
| Gambar 5.10 Fungsi <i>callback</i> untuk menangani data yang diterima dari <i>middleware</i> ..... | 23 |
| Gambar 5.11 Program <i>subscribe</i> untuk skenario pengiriman langsung.....                       | 23 |
| Gambar 5.12 Fungsi <i>callback</i> untuk menangani data yang diterima dari <i>middleware</i> ..... | 24 |
| Gambar 5.13 Program untuk melakukan koneksi internet .....   | 24 |
| Gambar 5.14 Program untuk memutus koneksi internet.....  | 25 |
| Gambar 5.15 Program untuk melakukan koneksi internet .....   | 25 |
| Gambar 5.16 Program untuk mengirim data dengan mekanisme penyimpanan sementara .....               | 25 |
| Gambar 5.17 Program untuk mengirim data dengan mekanisme pengiriman langsung .....                 | 26 |
| Gambar 6.1 Hasil pengujian <i>subscribe</i> ke semua topik .....                                   | 27 |
| Gambar 6.2 Hasil pengujian <i>subscribe</i> ke satu topik.....                                     | 28 |
| Gambar 6.3 Hasil pengujian penerimaan data JSON.....   | 29 |
| Gambar 6.4 Hasil pengujian pengiriman data secara berkala .....                                    | 30 |
| Gambar 6.5 Hasil pengujian pengiriman langsung.....  | 31 |
| Gambar 6.6 Hasil pengujian pengiriman dengan HTTP header .....                                     | 32 |
| Gambar 6.7 Grafik hasil pengujian <i>delay</i> dengan rentang waktu .....                          | 33 |
| Gambar 6.9 Grafik pengujian <i>throughput</i> pengiriman langsung .....                            | 34 |

## BAB 1 PENDAHULUAN

### 1.1 Latar belakang

*Internet of Things (IoT)* beberapa tahun terakhir ini, menjadi bagian penting dalam kehidupan, dan pendekatan yang populer dalam menyelesaikan masalah serta membangun aplikasi bisnis. Penyelesaian masalah dengan pendekatan *Internet of Things* dapat dilihat dari beberapa aplikasi IoT dalam manajemen bangunan seperti kontrol suhu, kelembaban, monitor aktivitas untuk manajemen penggunaan energi, dan kontrol *heating, ventilation* dan *air conditioning* (HVAC) (Gubbi et al., 2013).

Menurut Tan, IoT adalah sistem untuk menghubungkan berbagai macam objek fisik melalui skema pengalamatan unik, struktur yang menyerupai *internet* untuk mencapai satu tujuan tertentu (Tan, 2010). Sedangkan menurut Meng Ma, dalam penelitiannya, IoT dibagi menjadi 4 *layer*, yaitu *sensing layer*, *network layer*, *middleware layer* dan *application layer*. *Sensing layer* menurut penelitian tersebut merupakan *layer* yang bertugas sebagai penghubung antara dunia virtual dengan dunia nyata dengan dilibatkannya berbagai macam alat sebagai pengumpul informasi, seperti pembaca RFID, sensor infrared, GPS, kamera dan lain-lain. Sedangkan *network layer* adalah *layer* yang berfungsi untuk membawa data dari *sensing layer* ke *layer* berikutnya menggunakan pengalamatan IPv4, IPv6 yang berbasis internet maupun jaringan seluler. *Middleware layer* adalah *layer* integrasi dari berbagai macam fungsi seperti manajemen komunikasi, perangkat, pemrosesan data, penalaran semantik, keamanan dan *interfaces*, *application layer* adalah *layer* yang menyediakan layanan IoT dalam berbagai domain kepada pengguna (Ma, Wang dan Chu, 2013).

Sebelumnya, sebuah penelitian telah *Internet of Things* yang terdiri dari *node sensor*, *internet gateway device* dan media penyimpanan untuk mengamati lingkungan perkantoran. *Layer sensor* yang digunakan terdiri dari perangkat *microcontroller* dan raspberry pi dengan jenis sensor yang digunakan adalah sensor suhu dan kelembaban serta sensor kamera. Pada penelitian tersebut *middleware layer* berfungsi untuk mengumpulkan data dari *node sensor* dengan mekanisme *publish subscribe*, digunakan pula MongoDB dan GridFS untuk media penyimpanan serta IoT *Application* dalam bentuk aplikasi web sebagai *application layer*. Penggunaan NoSQL database MongoDB dimanfaatkan sebagai penyimpanan data terstruktur dengan format JSON dari sensor DHT11 dan DHT22 kemudian dikombinasikan dengan GridFS sebagai media penyimpanan data tidak terstruktur berupa gambar dari sensor kamera (Arganata, Pramukantoro dan Yahya, 2018).

Namun demikian, penelitian “Pengembangan Sistem Penyimpanan Data Berbasis MongoDB dan GridFS untuk Menyimpan Data yang Beragam dari *Node Sensor*” oleh Arganata masih menemui kendala dalam penerapannya karena masih berjalan pada jaringan lokal atau intranet dan belum terdapat komponen

internet sehingga belum dapat disebut sebagai sistem Internet of *Things*. Jika digunakan media penyimpanan yang berbasis *cloud* atau terletak pada jaringan internet, data dari *Middleware* tidak dapat diteruskan ke media penyimpanan tersebut. Oleh karena itu, dibutuhkan perangkat *internet gateway device* yang dapat meneruskan data dari *middleware* pada jaringan lokal ke media penyimpanan pada internet.

Penelitian lain oleh Zhu memiliki masalah yang sama dalam meneruskan data dari *wireless sensor network* di jaringan lokal *smart home* ke jaringan internet. *Gateway* yang dibangun pada penelitian tersebut berfungsi untuk memberikan akses dari jaringan lokal pada sistem *smart home* ke jaringan luar. Penempatan IoT *gateway* pada penelitian tersebut di antara *wireless sensor network* yang berfungsi sebagai *layer* sensor dan *layer* aplikasi yang terletak di jaringan luar. Data dari *layer* sensor kemudian akan diteruskan oleh IoT *gateway* menuju ke *layer* aplikasi, selain itu IoT *gateway* juga berfungsi untuk menerima perintah dari *layer* aplikasi untuk diteruskan ke *layer* sensor. Koneksi GPRS dan koneksi internet melalui *ethernet* digunakan untuk berkomunikasi dengan *layer* aplikasi, sedangkan komunikasi serial dengan modul serial *transceiver* digunakan untuk berkomunikasi dengan *layer* sensor (Zhu et al., 2010).

Dari permasalahan untuk meneruskan data dari *middleware* ke media penyimpanan yang muncul pada lingkungan *internet of things*, penulis mengusulkan penerapan *internet gateway device* untuk mengoleksi dan mengirimkan data ke media penyimpanan berbasis *cloud*. Pengembangan *internet gateway device* di penelitian ini menggunakan perangkat Raspberry Pi dengan tambahan modul GSM sebagai penyedia koneksi internet melalui GPRS. Mekanisme penerimaan data dari *middleware* menggunakan metode *publish subscribe* melalui protokol *websocket* yang telah dikembangkan pada penelitian sebelumnya, sedangkan untuk pengiriman data ke media penyimpanan digunakan protokol HTTP dengan metode HTTP POST. Selain itu, penulis mengusulkan dua skenario yang dapat digunakan. Skenario pertama data yang diterima selama rentang waktu tertentu ditampung terlebih dahulu dalam satu *file* JSON kemudian dikirimkan secara berkala secara bersamaan, sedangkan skenario kedua, setiap data yang diterima dari *middleware* akan langsung dikirimkan secara langsung ke media penyimpanan yang ada pada jaringan *internet*.

Pengembangan *internet gateway device* ini dilakukan dengan harapan agar data dari *middleware* dapat diteruskan ke media penyimpanan meskipun digunakan media penyimpanan yang berbasis *cloud* dan berada pada jaringan yang berbeda dengan *middleware*. Dengan diterapkannya *internet gateway device* untuk mengoleksi data dari *middleware* dan media penyimpanan di internet, komponen internet dapat ditambahkan pada lingkungan penelitian sebelumnya sehingga sistem tersebut dapat dikategorikan sebagai sistem *Internet of Things*.



## 1.2 Rumusan masalah

1. Bagaimana membangun sistem *internet gateway device* agar dapat menerima data dari *middleware* dan mengirim data ke media penyimpanan yang ada pada *cloud* menggunakan jaringan GPRS?
2. Bagaimana menerapkan sistem *internet gateway device* pada lingkungan IoT untuk menerima data dari *middleware* dan mengirim data ke media penyimpanan?
3. Bagaimana kinerja sistem *internet gateway device* dalam menerima data dari *middleware* dan mengirimkan data ke media penyimpanan pada *cloud* menggunakan jaringan GPRS?

## 1.3 Tujuan

Untuk memperjelas penelitian yang dikerjakan, maka ditentukan tujuannya sebagai berikut :

1. Dapat membangun sistem *internet gateway device* agar dapat menerima data dari *middleware* dan mengirim data ke media penyimpanan yang ada pada *cloud* menggunakan jaringan GPRS.
2. Dapat menerapkan sistem *internet gateway device* pada lingkungan IoT untuk menerima data dari *middleware* dan mengirim data ke media penyimpanan.
2. Dapat mengetahui kinerja sistem *internet gateway device* dalam menerima data dari *middleware* dan mengirimkan data ke media penyimpanan pada *cloud* menggunakan jaringan GPRS.

## 1.4 Manfaat

Manfaat dari penelitian ini adalah untuk membangun *sistem internet gateway device* yang dapat mengirimkan data ke media penyimpanan berbasis *cloud* menggunakan jaringan GPRS yang dapat digunakan sebagai rujukan dalam pengembangan sistem berbasis *Internet Of Things* serta menjadi acuan penelitian selanjutnya untuk mengembangkan beberapa aspek dalam penelitian ini yang masih dapat ditingkatkan lagi. Selain itu, bagi penulis penelitian ini bermanfaat untuk memahami lebih lanjut sistem *Internet Gateway Device* pada *Internet of Things*.

## 1.5 Batasan masalah

Agar permasalahan yang dirumuskan dapat lebih terfokus, maka pada penelitian ini dibatasi dalam hal :

1. Dalam pengembangan sistem *Internet Gateway Device* digunakan jaringan GPRS dari penyedia layanan Telkomsel dalam lingkup Fakultas Ilmu Komputer Universitas Brawijaya
2. Pengujian berfokus pada pengiriman data sensor ke media penyimpanan berbasis *cloud*

3. Data yang digunakan berupa JSON dari sensor temperatur dan sensor kelembaban DHT11/DHT22 serta gambar dari sensor kamera

## 1.6 Sistematika pembahasan

Sistematika penulisan skripsi ini sebagai berikut :

### BAB 1 Pendahuluan

Pada bab ini berisi latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, serta sistematika penulisan dari “Pengembangan Internet *Gateway Device* Berbasis GPRS Menggunakan Modul Gsm pada *Internet Of Things* untuk Mendukung Interkoneksi Jaringan Lokal Dengan Internet”.

### BAB 2 Landasan Kepustakaan

Pada bab ini berisi tentang penjelasan dan dasar-dasar teori beserta rujukan penelitian sebelumnya yang berhubungan dengan penelitian yang “Pengembangan Internet *Gateway Device* Berbasis GPRS Menggunakan Modul Gsm pada *Internet Of Things* untuk Mendukung Interkoneksi Jaringan Lokal Dengan Internet”.

### BAB 3 Metodologi Penelitian

Pada bab ini berisi tentang penjelasan dari langkah-langkah yang dilakukan dalam penelitian untuk membangun sistem Internet *Gateway Device* mulai dari studi literatur, analisis kebutuhan, perancangan sistem, implementasi, pengujian dan analisis serta pengambilan keputusan dan saran.

### BAB 4 Analisis Kebutuhan

Pada bab ini berisi analisis kebutuhan dari sistem Internet *Gateway Device* serta perancangan dari sistem yang kemudian akan diimplementasikan.

### BAB 5 Perancangan dan Implementasi

Pada bab ini berisi proses implementasi sistem Internet *Gateway Device* untuk memenuhi kebutuhan serta perancangan dari sistem.

### BAB 6 Pengujian dan Pembahasan Hasil Pengujian

Bab pengujian dan pembahasan hasil pengujian menjelaskan proses untuk menguji sistem Internet *Gateway Device* yang telah dibangun untuk memastikan apakah telah sesuai dengan kebutuhan yang disusun pada bab analisis kebutuhan dan perancangan.

### BAB 7 Penutup

Pada bab ini berisi ringkasan dan pencapaian hasil dan menjawab pertanyaan dari rumusan masalah. Serta saran yang dapat diberikan untuk pengembangan lebih lanjut.

## BAB 2 LANDASAN KEPUSTAKAAN

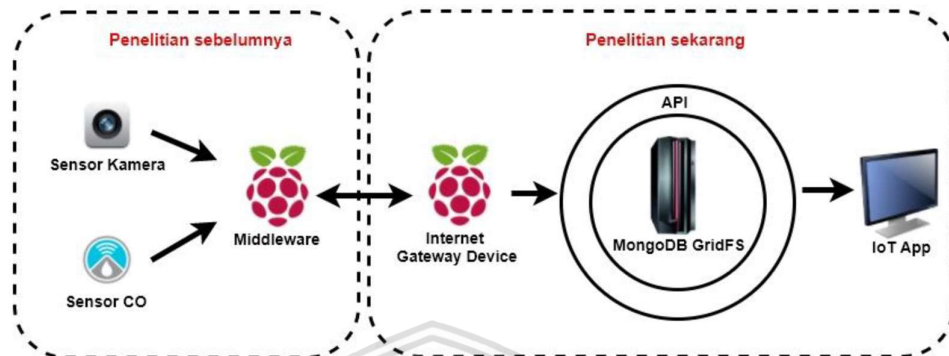
### 2.1 Kajian Pustaka

Kajian pustaka yang digunakan penulis pada penelitian ini antara lain penelitian berjudul “Pengembangan *IoT Middleware* Berbasis Event- Based Dengan Protokol Komunikasi Coap, Mqtt Dan *Websocket*” oleh Anwari dkk (2017). Penelitian tersebut menerapkan 3 protokol komunikasi yaitu coap, mqtt dan *websocket* untuk mengatasi permasalahan interoperabilitas pada *middleware internet of things*. Dua protokol yaitu coap dan mqtt masing-masing digunakan sebagai protokol komunikasi antara sensor dengan *middleware* sedangkan *websocket* digunakan sebagai protokol komunikasi antara *middleware* dengan *web app*. Selain menerapkan tiga protokol tersebut, pada penelitian ini juga menerapkan *redis* sebagai sistem penyimpanan sementara data dari sensor yang diterima oleh *middleware* yang bertindak sebagai *broker*, setiap sensor melakukan *publish* pada *middleware*, topik yang digunakan akan tersimpan pada *redis* sebagai *key*, dan data yang diterima sebagai *value* dari *key* tersebut.

Selain penelitian tersebut, penulis juga merujuk pada penelitian sebelumnya yang berjudul Pengembangan Sistem Penyimpanan Data Berbasis MongoDB dan GridFS untuk Menyimpan Data yang Beragam dari *Node Sensor* oleh Arganata dkk (2017) yang telah menerapkan sistem penyimpanan data pada IoT yang terpisah. Pada penelitian tersebut digunakan *internet gateway device* yang menerima data dari *middleware* menggunakan metode *publish subscribe*. Data yang diterima dari sensor CO dan sensor kelembaban berupa data terstruktur dalam format JSON, sedangkan data dari sensor kamera berupa data tidak terstruktur dalam bentuk gambar dengan format jpg yang kemudian data-data tersebut dikirimkan pada media penyimpanan MongoDB dan GridFS melalui sebuah RESTful *webservice*. Disebutkan pada penelitian ini bahwa data-data tersebut dikirimkan dari *Internet Gateway Device* ke RESTful *webservice* menggunakan metode HTTP POST selain itu RESTful *web service* pada penelitian tersebut juga digunakan dalam pembacaan data dari media penyimpanan MongoDB dan GridFS dengan menggunakan metode HTTP GET. Gambaran umum sistem pada penelitian ini dapat dilihat pada gambar 2.1

Penelitian lain yang menjadi rujukan bagi penulis pada penelitian ini yaitu penelitian dengan judul *IoT Gateway: Bridging Wireless Sensor Network into Internet of Things* (Zhu et al., 2010). Penelitian tersebut berfokus pada pembangunan *IoT gateway* yang berfungsi untuk memberikan akses ke jaringan luar pada jaringan lokal dalam sistem *smart home*. Pada penelitian tersebut *IoT Gateway* ditempatkan diantara *layer* aplikasi dan *layer* sensor yang berupa *wireless sensor network*. Selain berfungsi untuk menerima data dari sensor, dan menerima perintah dari *layer* aplikasi, *IoT Gateway* pada penelitian tersebut berfungsi untuk meneruskan data dari sensor menuju *layer* aplikasi. Untuk berkomunikasi dengan *layer* aplikasi, digunakan koneksi GPRS serta

koneksi internet melalui *Ethernet*, sedangkan untuk berkomunikasi dengan *node* sensor yang ada, digunakan komunikasi serial dengan menggunakan modul serial *transceiver*.



**Gambar 2.1** Gambaran sistem penyimpanan berbasis MongoDB dan GridFS

(Arganata, Pramukantoro dan Yahya, 2018)

## 2.2 Internet of Things

Menurut Lu Tan dan Neng Wang (2010), tidak ada definisi secara standar untuk *Internet of Things*. Dalam prosiding tersebut juga disebutkan bahwa secara fungsional *Internet of Things* dapat didefinisikan sebagai benda-benda yang memiliki identitas dan kepribadian yang beroperasi pada sebuah lingkungan yang cerdas untuk berkomunikasi dengan lingkungannya dalam konteks sosial dan dalam konteks pengguna.

## 2.3 Internet Gateway Device

*Internet Gateway Device* adalah sebuah perangkat yang dipakai untuk menghubungkan satu jaringan komputer dengan satu ataupun lebih jaringan komputer yang memakai protokol komunikasi yang berbeda sehingga informasi dari satu jaringan komputer bisa diberikan kepada jaringan komputer lain yang protokolnya tidak sama atau berbeda. Sama halnya dengan sebuah *gateway* jaringan, yang berfungsi sebagai sistem *internetworking* yang menghubungkan dua jaringan bersama-sama dan bisa dikonfigurasi dalam aplikasi perangkat lunak, perangkat keras, ataupun keduanya (Kende, et al., 2015)

## 2.4 Global System for Mobile communication (GSM)

GSM merupakan teknologi komunikasi seluler yang banyak diterapkan pada perangkat bergerak atau umumnya pada telepon genggam. Komunikasi GSM menggunakan sinyal digital yang dimodulasi berdasarkan waktu untuk mengirimkan informasi agar sampai ke penerima. Di seluruh dunia, GSM merupakan teknologi seluler yang paling banyak digunakan (Zikrillah, 2015)

## 2.5 General Packet Radio Service (GPRS)

GPRS telah lama diterapkan dalam layanan yang menggunakan konsep teknologi *Low Power Wide Area* (LWPA). GPRS menggunakan layanan radio yang berjalan diatas layanan GSM. Frekuensi gprs dibagi menjadi 4.6ms yang kemudian dibagi lagi menjadi 8 *timeslots*. *Frequency reuse* hingga 12 *scheme* dibutuhkan dalam GPRS sehingga menyebabkan *spectral density* yang tidak begitu efisien. Lebar pita yang digunakan GPRS memungkinkan GPRS untuk beroperasi tanpa tergantung pada *duty cycle* atau batasan *listen before talk* (Vejlgaard et al., 2017).

## 2.6 JavaScript Object Notation (JSON)

Dikutip dari *website* resmi JSON, JavaScript Object Notation (JSON) merupakan format pertukaran data yang diklaim memudahkan manusia dalam menulis dan membaca data dan memudahkan mesin dalam menerjemahkan dan menghasilkan data itu sendiri. Selain itu, JSON diperuntukkan untuk mempermudah pertukaran data antar bahasa pemrograman yang berbeda. Menurut standar yang dipublikasi pada ECMA internasional (2017) JSON memiliki syntax yang terdiri dari tanda kurung kurawal, tanda kurung, titik dua dan koma (ECMA, 2017).

## 2.7 Throughput

End-to-end *throughput* atau singkatnya *throughput* merupakan salah satu ukuran kinerja yang kritikal dalam jaringan komputer. Ketika dua komputer melakukan transfer *file* melalui sebuah jaringan komputer, jika *file* tersebut berukuran  $F$  bits dan memakan waktu selama  $T$  detik, maka *throughput* dari proses transfer tersebut adalah  $F/T$  bits/detik (Kurose, James dan Ross, 2013)

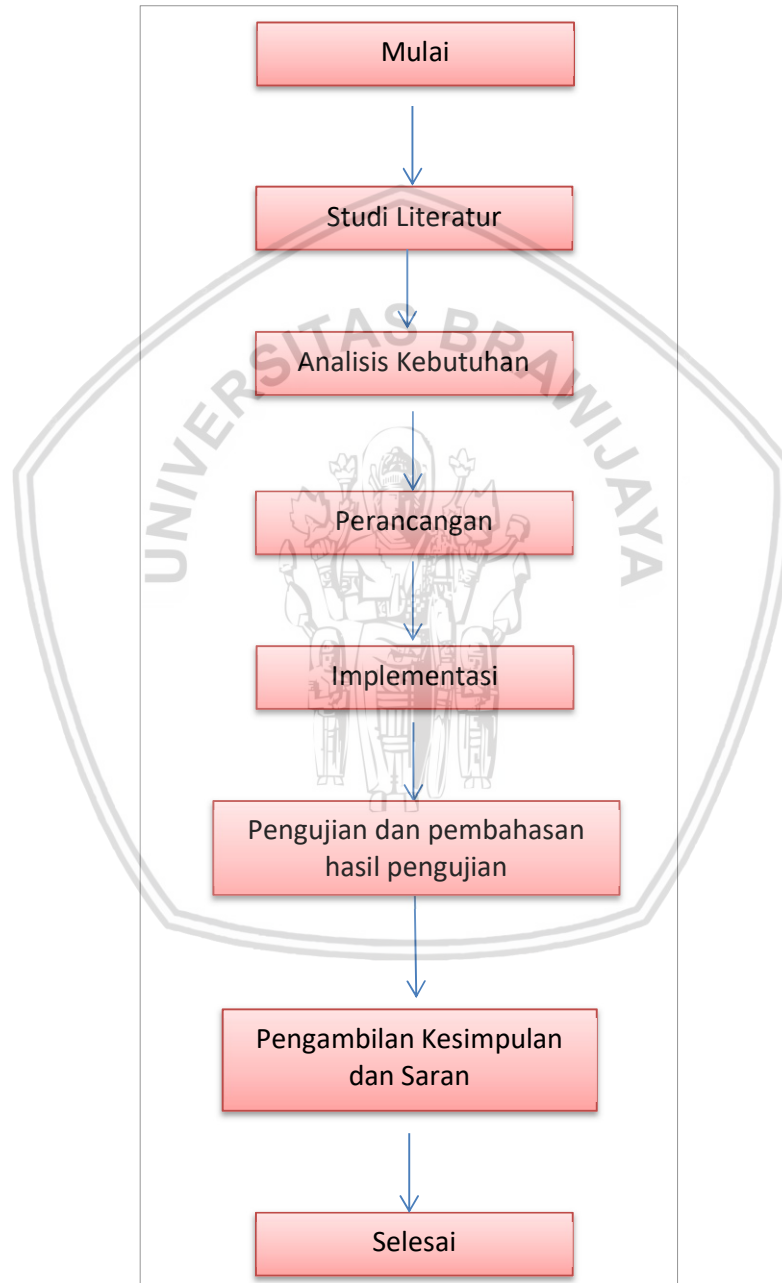
## 2.8 Delay

*Delay* merupakan parameter intrinsik pada komunikasi, informasi akan memakan waktu untuk mencapai *end point* dikarenakan jarak yang cukup jauh. Waktu *delay* dapat meningkat jika paket menghadapi antrian panjang dalam jaringan (*congestion*), atau melewati rute yang tidak langsung untuk menghindari *congestion* (Gouveia and Magedanz, 2011).



### BAB 3 METODOLOGI

Pada bab ini akan menjelaskan tentang tahapan-tahapan yang akan dilakukan dalam pengimplementasian, pengujian, dan analisis dari metode yang digunakan. Metodologi penelitian skripsi ini akan dikerjakan dalam beberapa tahap. Gambar 3.1 akan menunjukkan tahapan-tahapan dalam metodologi.



**Gambar 3.1** Tahap Pengerjaan Penelitian

### 3.1 Studi Literatur

Untuk menunjang penulisan skripsi Pengembangan *Internet Gateway Device* Berbasis GPRS Menggunakan Modul Gsm pada *Internet Of Things* untuk Mendukung Interkoneksi Jaringan Lokal Dengan Internet diperlukan studi literatur untuk menunjang dasar-dasar teori dan kajian pustaka antara lain

1. Penerimaan dan pengiriman data pada *Internet Gateway Device*
2. Pengiriman data menggunakan koneksi GPRS

Teori dan pustaka yang berkaitan diperoleh dari buku, jurnal, *e-book* dan dokumentasi pada penelitian sebelumnya.

### 3.2 Analisis Kebutuhan

Pada tahap analisis kebutuhan, penulis menganalisa sumber daya dari lingkungan *Internet of Things* yang telah ada serta sumber daya yang akan dibutuhkan dalam membangun sistem *Internet Gateway Device*. Kebutuhan tersebut meliputi kebutuhan jaringan yang digunakan dalam membangun sistem *Internet Gateway Device*, kebutuhan data untuk diterima dari *middleware* dan dikirimkan ke media penyimpanan, kebutuhan secara fungsional serta kebutuhan nonfungsional yang terdiri dari faktor *reliability* serta penggunaan data dari *Internet Gateway Device*.

### 3.3 Perancangan

Pada tahap perancangan, penulis membuat rancangan sistem *Internet Gateway Device* yang terdiri dari:

1. Perancangan dan pemodelan alur sistem secara keseluruhan.
2. Perancangan untuk pengalamatan dan arsitektur jaringan yang digunakan. Selain itu dilakukan juga .
3. Perancangan arsitektur jaringan secara keseluruhan, dan juga perancangan untuk melakukan pengujian sistem *internet gateway device* secara fungsional maupun nonfungsional.

### 3.4 Implementasi

Tahap implementasi dilakukan pada sistem dengan melakukan konfigurasi pada raspberry pi agar dapat terkoneksi ke *internet* menggunakan modul sim 808, implementasi program pada *Internet Gateway Device* untuk menerima data dari sensor serta mengirimkan data pada Media penyimpanan melalui *web service* API.

### 3.5 Pengujian dan Pembahasan Hasil Pengujian

Pada tahap ini dilakukan pengujian dan analisis terhadap sistem yang telah diterapkan. Pengujian dilakukan bertujuan untuk mengetahui apakah rancangan sistem *Internet Gateway Device* yang digunakan sudah memenuhi kebutuhan yang telah didefinisikan sebelumnya. Pengujian dilakukan menggunakan 2 jenis pengujian yaitu, pengujian fungsional, serta pengujian nonfungsional. Pengujian fungsional dilakukan untuk memastikan apakah kebutuhan fungsional yang telah didefinisikan sebelumnya sudah tercapai. Setelah kebutuhan fungsional tercapai seluruhnya maka dilanjutkan pada tahap pengujian selanjutnya yaitu pengujian nonfungsional yang terdiri dari pengujian *delay* dan *throughput*, *reliability* dan *scalability*.

### 3.6 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan dilakukan setelah tahapan pengujian sistem *Internet Gateway Device* dan analisis selesai dilakukan. Kesimpulan diambil dari hasil pengujian dan analisis sistem *Internet Gateway Device*. Pengambilan saran menjadi tahap terakhir yang dilakukan untuk mengevaluasi adanya kesalahan-kesalahan dan menyempurnakan penulisan serta sebagai pertimbangan atas perkembangan sistem *Internet Gateway Device* lebih lanjut.

## BAB 4 ANALISIS KEBUTUHAN

Analisis kebutuhan dan perancangan pada penelitian ini diperlukan untuk memastikan pengembangan sistem *Internet Gateway Device* berjalan secara sistematis. Analisis kebutuhan berisi sumber daya yang dibutuhkan dalam mengembangkan sistem *Internet Gateway Device* untuk mengirimkan data ke media penyimpanan berbasis *cloud* menggunakan jaringan GPRS.

### 4.1 Deskripsi Umum Sistem

Penelitian ini akan berfokus pada *Internet Gateway Devices* yang bertugas menerima data dari *middleware* menggunakan mekanisme *publish* dan *subscribe* melalui protokol *websocket*, data yang diterima berupa data dalam format JSON. Selain itu data yang telah diterima akan dikirimkan ke media penyimpanan berupa MongoDB dan GridFS berbasis *cloud* melalui *webservice* / API. Pengiriman data menuju *cloud* menggunakan jaringan GPRS dari modul GSM yang terpasang pada alat Raspberry Pi.

### 4.2 Batasan

1. Perangkat lunak untuk menerima data menggunakan bahasa pemrograman python dan berjalan pada python versi 2
2. Perangkat lunak untuk melakukan koneksi GPRS berjalan dengan perangkat lunak ppp

### 4.3 Lingkungan Operasi

Lingkungan operasi yang digunakan oleh perangkat *Internet Gateway* adalah

- |             |  |
|-------------|--|
| 1. OS       | : Raspbian 9 Stretch                     |
| 2. Prosesor | : 900 MHz 32-bit quad-core ARM Cortex-A7 |
| 3. RAM      | : 1GB                                    |
| 4. Storage  | : 16GB atau lebih                        |

### 4.4 Kebutuhan Jaringan

Kebutuhan jaringan mendeskripsikan kebutuhan dalam membangun sistem media penyimpanan data IoT yang berupa kebutuhan jaringan atau hubungan antar perangkat. Kebutuhan jaringan pada penelitian ini adalah *Internet Gateway Device* terkoneksi ke media penyimpanan MongoDB dan GridFS berbasis *cloud* menggunakan koneksi GPRS.

## 4.5 Kebutuhan Data

Kebutuhan data merupakan deskripsi dari data yang dibutuhkan oleh sistem internet *gateway device*. Data yang dipakai merupakan data sensor DHT11 dan DHT22 serta data dari sensor kamera yang diterima melalui *middleware*. Salah satu contoh dari *payload* data sensor DHT11 dan DHT22 dapat dilihat pada tabel 4.1.

**Tabel 4.1 Payload data sensor DHT11 dan DHT22**

```
{ "protocol": "mqtt", "temperature": { "unit": "celcius", "value": "25.0" }, "timestamp": 1524105130, "humidity": { "unit": "%", "value": "34.0" }, "topic": "office/roomA13", "sensor": { "index": 8456747, "tipe": "esp8266", "module": "dht11", "ip": "192.168.42.50" }}
```

Data yang dikirimkan oleh sensor DHT11 dan DHT22 terdiri dari jenis protokol yang digunakan, nilai suhu dan unit yang digunakan, *timestamp* dalam format epoch, nilai kelembaban dan unit yang digunakan, topik, indeks sensor, tipe mikrokontroler yang digunakan *node* sensor, modul sensor yang digunakan serta alamat IP yang dimiliki oleh *node* sensor.

Sedangkan untuk data dari sensor kamera, skema yang digunakan dapat dilihat pada tabel 4.2

**Tabel 4.2 Skema payload data sensor kamera**

```
{ "Data" : data, "Name" : "images0.jpg" }
```

*Payload* yang dikirimkan oleh sensor kamera berisikan data gambar dalam bentuk object pickle dengan format string serta nama gambar dengan ekstensi .jpg. *Payload* data sensor DHT11 dan DHT22 serta sensor kamera dikirimkan oleh *node* sensor ke *middleware* dalam format JSON.

## 4.6 Kebutuhan Sistem

Kebutuhan sistem mendeskripsikan kebutuhan yang diperlukan dalam membangun sistem *Internet Gateway Device*. Kebutuhan yang dimaksud meliputi kebutuhan fungsional dan kebutuhan nonfungsional.

### 4.6.1 Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan-kebutuhan yang harus ada dalam sistem internet *gateway device* yang dibangun, kebutuhan fungsional dibuat berdasarkan fungsi-fungsi yang dapat dilakukan oleh *internet gateway device*. Pada penelitian ini, kebutuhan fungsional yang harus dimiliki *internet gateway device* dapat dilihat pada Tabel 4.3.



**Tabel 4.3 Kebutuhan Fungsional Sistem**

| No. | Kebutuhan Fungsional  |
|-----|---|
| 1   | <i>Internet Gateway Device</i> dapat melakukan <i>subscribe</i> topik ke ke <i>middleware</i>   |
| 2   | <i>Internet Gateway Device</i> dapat melakukan <i>subscribe</i> salah satu topik dari <i>middleware</i>   |
| 3   | <i>Internet Gateway Device</i> dapat menerima data dari <i>middleware</i> dalam format JSON   |
| 4   | <i>Internet Gateway Device</i> dapat menyimpan sementara <i>file</i> yang diterima dari <i>middleware</i> dalam <i>file</i> JSON lalu mengirimkan ke media penyimpanan secara berkala |
| 5   | <i>Internet Gateway Device</i> dapat mengirimkan data yang diterima dari <i>middleware</i> ke media penyimpanan secara langsung   |
| 6   | <i>Internet Gateway Device</i> dapat mengirimkan data yang diterima dari <i>middleware</i> menggunakan HTTP <i>header</i>   |

#### 4.6.2 Kebutuhan Nonfungsional

Kebutuhan nonfungsional adalah kebutuhan yang mendukung sistem *internet gateway device* yang dibangun serta kebutuhan berupa kualitas dari sistem *internet gateway device* yang dibangun. Dalam penelitian ini, kebutuhan nonfungsional *internet gateway device* dapat dilihat pada Tabel 4.4.

**Tabel 4.4 Kebutuhan Nonfungsional Sistem**

| No. | Kebutuhan nonfungsional                      | Deskripsi   |
|-----|--|---|
| 1   | Pengujian <i>delay</i> dan <i>throughput</i> | Dapat diketahui nilai <i>delay</i> dan <i>throughput</i> saat pengiriman data menggunakan koneksi GPRS              |
| 2   | <i>Reliability</i>                           | <i>Internet gateway device</i> dapat mengirimkan data sejumlah 10, 20, 30, 40 dan 50 data untuk masing-masing topik |
| 3   | <i>Scalability</i>                           | <i>Internet gateway device</i> dapat mengirimkan data dengan ukuran 1MB, 2MB, 5MB dan 10MB                          |

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini dilakukan perancangan sistem *internet gateway device* dan penerapan terhadap perancangan yang telah didefinisikan. Hasil dari implementasi kemudian akan dipakai untuk mengukur keberhasilan pengembangan sistem media penyimpanan yang dilakukan pada penelitian ini. Tahap implementasi akan dilakukan berdasarkan perancangan, yang meliputi implementasi jaringan, implementasi *Internet Gateway Device*, implementasi *web service* dan Implementasi media penyimpanan.

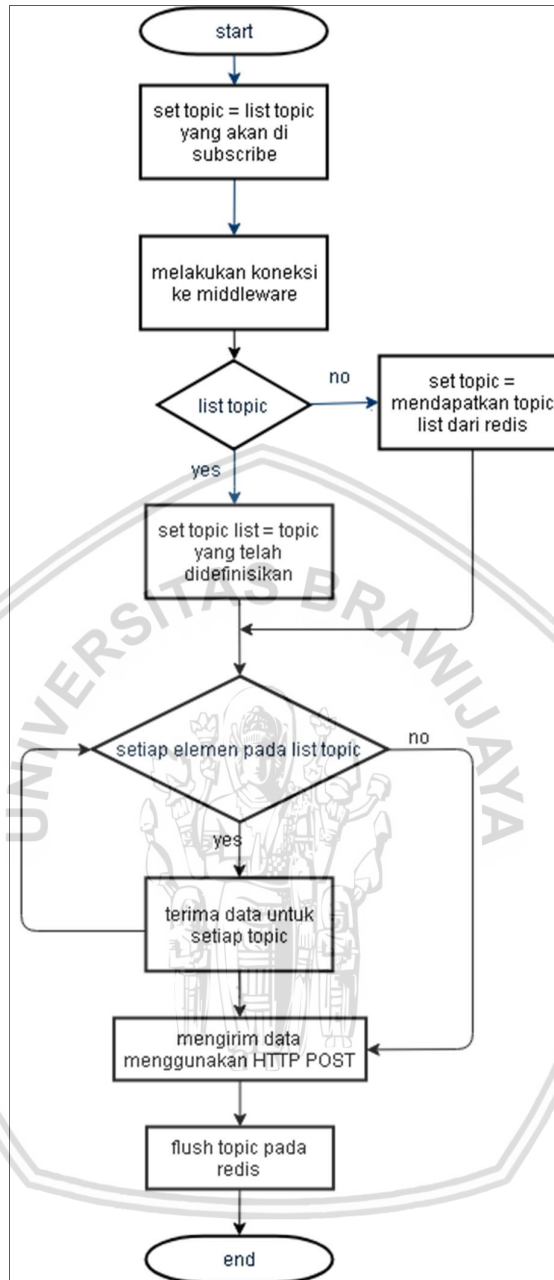
### 5.1 Perancangan

Proses perancangan pada penelitian ini dibutuhkan agar proses implementasi lebih mudah dan lebih terarah dengan disesuaikan dengan acuan yang telah dideskripsikan pada proses perancangan. Perancangan pada penelitian ini dibagi menjadi perancangan alur sistem berupa gambaran umum bagaimana keseluruhan sistem berkomunikasi, perancangan pengalamatan dan perancangan arsitektur jaringan, selain itu juga dilakukan perancangan pengujian yang akan dilakukan.

#### 5.1.1 Perancangan Alur Sistem

Rancangan dari alur sistem diperlukan agar sistem dari *internet gateway device* dapat berjalan dengan sebagaimana mestinya. Perancangan alur sistem menggambarkan bagaimana alur data mulai dari *middleware* hingga dapat disimpan pada media penyimpanan. Rancangan alur sistem pada penelitian ini dibagi berdasarkan dua skenario yang digunakan, pada skenario pertama, digunakan rentang waktu tertentu untuk melakukan *subscribe* ke *middleware* sedangkan data yang diterima akan disimpan sementara pada *file JSON* sebelum dikirim bersamaan ke media penyimpanan setelah waktu *subscribe* selesai. Skenario yang kedua digunakan skema pengiriman langsung ke media penyimpanan setelah data dari *middleware* diterima sehingga kondisi koneksi ke internet terus menyala.

Secara umum mekanisme yang digunakan dapat dilihat pada gambar 5.1. *Internet gateway device* sebelumnya akan menentukan topik yang akan dilakukan *subscribe*, setelah itu dilakukan koneksi ke *middleware* melalui *interface wireless LAN* untuk mendapatkan data yang ada di *middleware*. Setelah melakukan *subscribe* pada satu atau beberapa topik, data yang didapatkan akan dikirimkan ke media penyimpanan melalui protokol HTTP dengan metode POST ke alamat <http://iot.dijalinbersari.com:5001/api/post>. Pengiriman dilakukan dengan koneksi GPRS yang didukung dari modul GSM melalui *interface ppp0*. Setelah pengiriman selesai, dilakukan *flush* atau penghapusan data pada *redis*, hal ini dilakukan agar data yang telah tersimpan pada media penyimpanan tidak dikirimkan kembali oleh *middleware* sehingga tidak terjadi penumpukan data yang sama pada media penyimpanan.

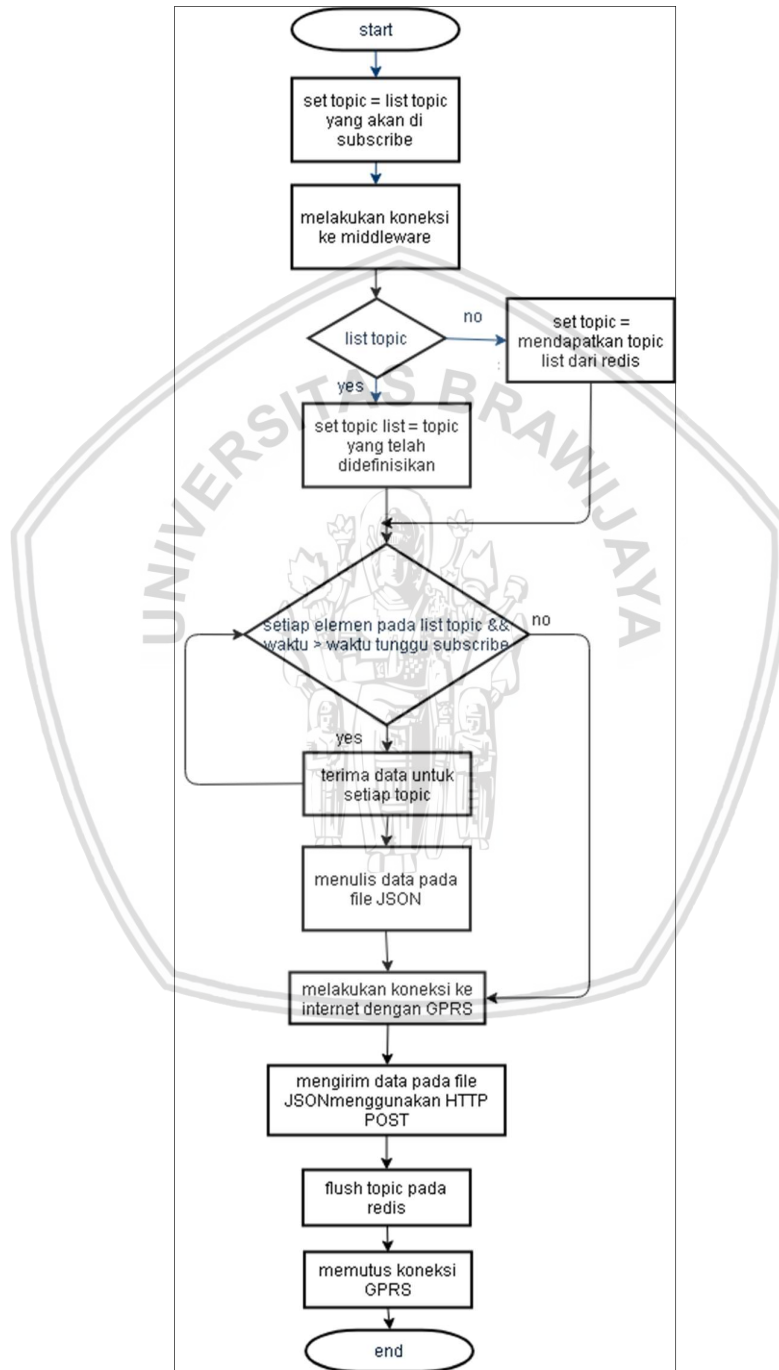


Gambar 5.1 Diagram alir sistem secara umum

#### 5.1.1.1 Alur skenario dengan rentang waktu subscribe

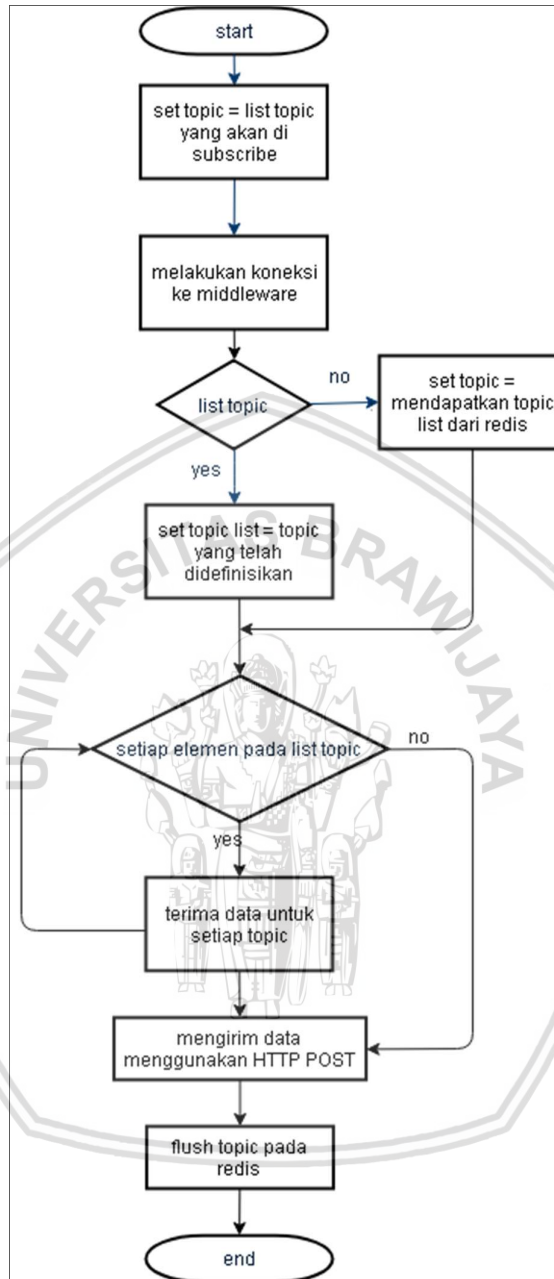
Pada skenario pertama, *internet gateway device* akan melakukan *subscribe* ke *middleware* dalam rentang waktu tertentu, selama melakukan *subscribe* tersebut, setiap data yang diterima dari *middleware* akan disimpan sementara pada *file* dengan format JSON pada *internet gateway device*. *Topic* yang akan dilakukan *subscribe* oleh *internet gateway device* didapatkan dari *redis* atau dari input yang diberikan ketika menjalankan program. Saat waktu untuk *subscribe* telah terlampaui maka semua data yang tersimpan pada *file*

JSON akan dikirimkan ke media penyimpanan menggunakan protokol HTTP melalui koneksi GPRS. Setelah data telah terkirim ke media penyimpanan, akan dilakukan *flush database redis* pada *middleware* untuk memastikan bahwa data yang telah tersimpan pada media penyimpanan tidak diterima kembali ketika melakukan *subscribe*. Alur skenario pertama dapat dilihat pada gambar 5.2



Gambar 5.2 Rancangan alur skenario pertama

### 5.1.1.2 Alur skenario pengiriman langsung



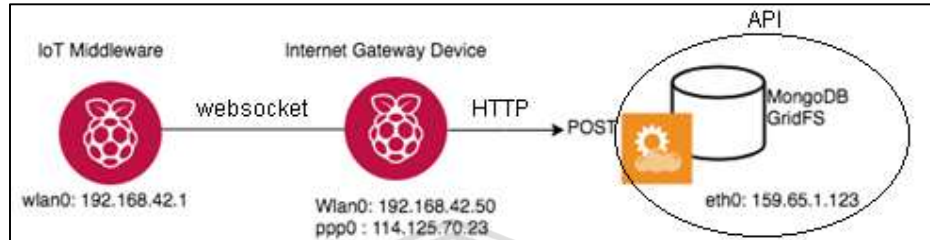
**Gambar 5.3 Rancangan alur skenario kedua**

Pada skenario pengiriman langsung, setiap data yang diterima oleh internet *gateway device* dari *middleware* akan langsung dikirimkan ke media penyimpanan dengan menggunakan protokol HTTP melalui koneksi GPRS dengan kondisi koneksi GPRS selalu menyala sehingga tidak perlu dilakukan pembentukan koneksi terlebih dahulu dengan GPRS sebelum dilakukan pengiriman data. *Flush database redis* juga diperlukan seperti pada skenario pertama, untuk memastikan data yang telah tersimpan pada media

penyimpanan tidak diterima lagi ketika melakukan *subscribe*. Alur skenario kedua yang digunakan dapat dilihat pada gambar 5.3

### 5.1.2 Perancangan Pengalaman dan Arsitektur Jaringan

Perancangan pengalaman serta arsitektur jaringan yang digunakan pada penelitian ini dimodelkan seperti pada gambar 5.4



**Gambar 5.4 Arsitektur jaringan yang digunakan pada penelitian ini**

Perangkat utama yang digunakan pada penelitian ini adalah internet *gateway device* yang menggunakan perangkat raspberry pi dengan tambahan modul GSM. Interface wlan0 pada perangkat Internet *gateway device* digunakan untuk menerima data dari *middleware*, sedangkan untuk mengirim data ke media penyimpanan yang berbasis *cloud*, digunakan koneksi GPRS melalui interface ppp0.

Internet *gateway device* akan melakukan *subscribe* ke *middleware* untuk menerima data menggunakan protokol *websocket* pada port 3000. Data yang diterima dari *middleware* akan diteruskan ke media penyimpanan menggunakan protokol HTTP dengan metode POST pada port 5001.

### 5.1.3 Perancangan Pengujian

Perancangan pengujian dibutuhkan sebagai skenario yang akan dilakukan untuk melakukan pengujian yang kemudian berfungsi sebagai alat untuk memvalidasi kebutuhan-kebutuhan yang telah didefinisikan.

#### 5.1.3.1 Pengujian fungsional

Pengujian fungsional dilakukan untuk mengetahui apakah kebutuhan fungsional sistem media penyimpanan yang sebelumnya didefinisikan telah terpenuhi. Skenario yang akan dilakukan dalam tahap pengujian fungsional sistem media penyimpanan penelitian ini dapat dilihat pada tabel 5.1

**Tabel 5.1 Skenario pengujian fungsional**

| No. | Kebutuhan Fungsional  | Skenario Pengujian  |
|-----|---|---|
| 1   | Internet <i>Gateway Device</i> dapat mensubscribe semua <i>topic</i> dari <i>middleware</i> | <ol style="list-style-type: none"> <li>1. <i>Middleware</i> sudah berjalan</li> <li>2. Pengguna menjalankan program berbasis python untuk melakukan susbcribe pada <i>middleware</i> dengan <i>topic</i> yang didapatkan dari <i>redis key</i></li> </ol> |



| No. | Kebutuhan Fungsional   | Skenario Pengujian   |
|-----|--|--|
| 2   | Internet Gateway Device dapat mensubscribe satu topic dari middleware  | <ol style="list-style-type: none"> <li>1. <i>Middleware</i> sudah berjalan</li> <li>2. Pengguna menjalankan program berbasis python untuk melakukan <i>subscribe</i> pada <i>middleware</i> dengan <i>topic</i> yang ditentukan dari parameter saat menjalankan program</li> </ol>               |
| 3   | Internet Gateway Device dapat menerima data dari <i>middleware</i> dalam format JSON   | <ol style="list-style-type: none"> <li>1. <i>Middleware</i> sudah berjalan</li> <li>2. Pengguna menjalankan program berbasis python untuk menerima data JSON dengan melakukan <i>subscribe</i> pada <i>middleware</i> dengan <i>topic</i> yang diperoleh dari <i>redis key</i></li> </ol>        |
| 4   | Internet Gateway Device dapat menyimpan sementara <i>file</i> yang diterima dari <i>middleware</i> dalam <i>file</i> JSON lalu mengirimkan ke media penyimpanan secara berkala | <ol style="list-style-type: none"> <li>1. <i>Middleware</i> sudah berjalan</li> <li>2. Pengguna menjalankan program berbasis python untuk menerima data <i>File</i> dengan melakukan <i>subscribe</i> pada <i>middleware</i> dengan <i>topic</i> yang diperoleh dari <i>redis key</i></li> </ol> |
| 5   | Internet Gateway Device dapat mengirimkan data yang diterima dari <i>middleware</i> ke media penyimpanan secara langsung   | <ol style="list-style-type: none"> <li>1. <i>Middleware</i> sudah berjalan</li> <li>2. Pengguna menjalankan program berbasis python untuk menerima data <i>File</i> dengan melakukan <i>subscribe</i> pada <i>middleware</i> dengan <i>topic</i> yang diperoleh dari <i>redis key</i></li> </ol> |

| No. | Kebutuhan Fungsional  | Skenario Pengujian   |
|-----|---|--|
| 6   | Internet Gateway Device dapat mengirimkan data yang diterima dari <i>middleware</i> menggunakan HTTP header | <ol style="list-style-type: none"> <li>1. <i>Middleware</i> sudah berjalan</li> <li>2. Pengguna menjalankan program berbasis python untuk menerima data <i>File</i> dengan melakukan <i>subscribe</i> pada <i>middleware</i> dengan <i>topic</i> yang diperoleh dari <i>redis key</i></li> </ol> |

### 5.1.3.2 Pengujian nonfungsional

Pengujian nonfungsional dilakukan untuk memastikan bahwa kebutuhan nonfungsional untuk media penyimpanan dapat terpenuhi sepenuhnya. Beberapa pengujian nonfungsional yang dilakukan pada penelitian ini dapat dilihat pada tabel 5.2. Pengujian *delay* dan *throughput* dilakukan dengan cara menjalankan program untuk menerima dan mengirimkan data, data yang diolah pada pengujian *delay* dan *throughput* adalah ukuran data yang diterima dan dikirim serta lama waktu untuk mengirimkan data tersebut ke media penyimpanan. Data ukuran serta lama waktu didapatkan dengan menampilkannya langsung dengan program yang digunakan untuk menerima dan mengirimkan data. Pengujian penggunaan data dilakukan dengan menjalankan program yang digunakan untuk menerima dan mengirimkan data secara terus menerus selama 24 jam kemudian mengamati penggunaan data melalui program *vncstat* yang terinstall pada raspberry pi. Pengujian *reliability* dilakukan dengan cara mengirimkan data JSON dengan ukuran 1MB, 2MB, 5MB dan 10MB kemudian mengamati dengan menggunakan *wireshark*. apakah data berhasil terkirim ke media penyimpanan.

**Tabel 5.2 Deskripsi pengujian nonfungsional**

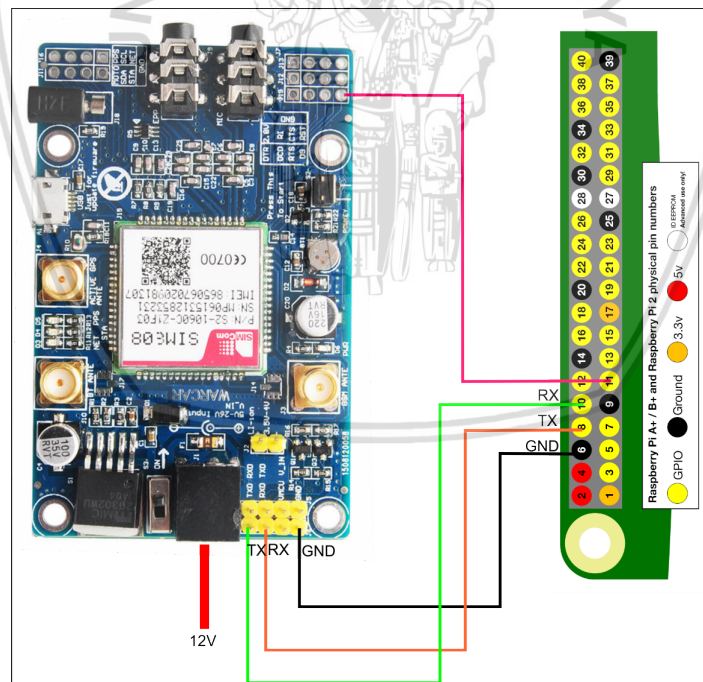
| No. | Pengujian nonfungsional                      | Skenario Pengujian  |
|-----|--|---|
| 1   | Pengujian <i>delay</i> dan <i>throughput</i> | Menjalankan skenario pertama dan kedua kemudian menghitung <i>delay</i> dan <i>throughput</i> pengiriman ke media penyimpanan |
| 2   | <i>Reliability</i>                           | Melakukan <i>subscribe</i> dan mengirimkan data dengan jumlah 10, 20, 30, 40 dan 50 data untuk masing-masing topik            |
| 3   | <i>Scalability</i>                           | Mengirimkan data dengan ukuran 1MB, 2MB, 5MB dan 10MB   |

## 5.2 Implementasi

Tahap implementasi dilakukan berdasarkan dari perancangan yang telah dilakukan sebelumnya. Tahap implementasi, dibagi menjadi 3 bagian yaitu implementasi untuk menghubungkan internet *gateway device* dengan *middleware* dan media penyimpanan yang dijelaskan pada implementasi jaringan, implementasi untuk menerima data dari *middleware* yang dijelaskan pada implementasi *subscriber* dan implementasi pengiriman dengan HTTP POST untuk mengimplementasikan pengiriman data ke media penyimpanan.

### 5.2.1 Implementasi jaringan

Implementasi jaringan dilakukan dengan menghubungkan Internet *Gateway Device* ke Internet menggunakan modul SIM 808 agar dapat melakukan POST ke *webservice*. Untuk menghubungkan *Gateway Device* dengan internet dilakukan menggunakan modul SIM 808 dengan jaringan GPRS. Untuk menghubungkan raspberry pi dengan modul sim 808 digunakan skema pada gambar 10. Agar dapat berfungsi, pin TX pada modul sim 808 dihubungkan dengan pin 10 GPIO yang difungsikan sebagai pin RX, pin RX pada modul sim 808 dihubungkan dengan pin 8 GPIO yang berfungsi sebagai pin TX, dan pin ground pada sim 808 dihubungkan pada pin 6 yang berfungsi sebagai pin ground.



**Gambar 5.5 Skema Koneksi GPIO dan modul sim 808**

Untuk menyalakan modul sim 808 digunakan *power supply* yang terpisah dari raspberry pi yang memberikan *power* sebesar 12v 1 Ampere. Skema untuk

menghubungkan modul GSM dengan perangkat raspberry pi dapat dilihat pada gambar 5.5

Agar modul sim 808 dapat menyala kembali secara otomatis setelah dilakukan *reboot* pada raspberry pi maka perlu dilakukan konfigurasi tambahan dengan menghubungkan pin D9 pada modul sim 808 dengan salah satu pin digital pada GPIO, dengan demikian perintah untuk menyalakan modul sim 808 dapat dilakukan dengan memberikan sinyal digital melalui GPIO menggunakan program berbasis python seperti pada gambar 5.6

|  |                                  |
|--|----------------------------------|
|  |                                  |
|  | DO send digital signal to pin 11 |
|  | DO sleep for 4 seconds           |

**Gambar 5.6 Pseudocode untuk menyalakan modul SIM 808**

Selain itu, agar modul sim 808 dapat menyala setiap kali perangkat dilakukan *reboot*, maka program yang digunakan untuk menyalakan modul sim 808 perlu ditambahkan pada cronjob seperti pada gambar 5.7

```
# m h dom mon dow   command
@reboot /home/pi/powerOn808.py
```

**Gambar 5.7 Konfigurasi cronjob pada perangkat Internet Gateway**

### 5.2.2 Implementasi *subscriber*

Agar internet *gateway device* dapat menerima data yang dikirimkan dari *middleware* diperlukan proses *subscribe* ke alamat *middleware* dengan menggunakan protokol *websocket* pada port 3000. *Topic* yang akan *unsubscribe* oleh internet *gateway device* merujuk pada *topic* yang tersimpan pada *middleware* dalam bentuk *redis key*. Internet *gateway device* kemudian dapat mengambil semua *topic* yang ada pada *redis* untuk *unsubscribe*, selain itu *topic* juga dapat diberikan melalui argumen ketika menjalankan program *subscribe*. Mekanisme penentuan *topic* dapat dilihat pada gambar 5.8

|   |  |
|---|--|
| <i>Pseudocode</i> fungsi menentukan topik |  |
|   | DEF <i>getTopic()</i> function                         |
|   | IF input <i>topic</i>                                  |
|   | SET <i>topics</i> = input <i>topic</i>                 |
|   | ELSE   |
|   | SET <i>topics</i> = get <i>topic</i> from <i>redis</i> |
|   | RETURN <i>topics</i>                                   |
|   | END  |

**Gambar 5.8 Pseudocode untuk menentukan *topic***

Implementasi dari *subscriber* kemudian akan dibagi menjadi dua, berdasarkan skenario yang diajukan.

#### 5.2.2.1 Skenario dengan rentang waktu *subscribe*

Skenario pertama untuk *subscribe* diimplementasikan dengan memberikan rentang waktu dalam menerima data dari *middleware*, dan ketika data

diterima, data akan disimpan sementara pada sebuah *file* JSON. Implementasi *subscriber* untuk skenario pertama dapat dilihat pada gambar 5.9

| Pseudocode <i>subscribe</i> dengan rentang waktu |  |
|--|--|
|  | <pre> SET topic_list = getTopic() SET socket_client = socket(192.168.42.1,3000) FOR each topic in topic_list     ON socket_client.subscribe DO on_response()     DO socket_client.emit('subscribe',topic) DO wait data for 1800 seconds DO sendHTTP() FOR each topic in topic_list     DO flush topic END </pre> |

**Gambar 5.9 Pseudocode program *subscribe* dengan skenario waktu tunggu**

Proses penyimpanan data yang diterima dari *middleware* ke *file* JSON didefinisikan pada fungsi *callback* ketika data diterima pada saat proses *subscribe* berjalan. Fungsi *callback* yang digunakan untuk menyimpan data sementara dapat dilihat pada gambar 5.10

| Pseudocode fungsi untuk menangani data |  |
|--|--|
|  | <pre> DEFINE on_response()     DO dump received data to JSON file END </pre> |

**Gambar 5.10 Pseudocode fungsi *callback* untuk menangani data yang diterima dari *middleware***

#### 5.2.2.2 Skenario pengiriman langsung

Skenario kedua untuk *subscribe* diimplementasikan dengan mengirimkan langsung data dari *middleware* setelah data tersebut diterima sehingga tidak diperlukan *file* JSON sebagai penyimpanan sementara data yang diterima. Implementasi proses *subscribe* untuk skenario pengiriman langsung dapat dilihat pada gambar 5.11

| Pseudocode <i>subscribe</i> pengiriman langsung |  |
|---|--|
|   | <pre> SET topics = getTopic() SET socket_client = socket(192.168.42.1,3000) FOR each topic in topic_list     ON socket_client.subscribe DO on_response()     DO socket_client.emit('subscribe',topic) END </pre> |

**Gambar 5.11 Pseudocode program *subscribe* untuk skenario pengiriman langsung**

Pada skenario kedua, fungsi *callback* digunakan untuk proses pengiriman data dan penghapusan data dari *topic* yang telah terkirim ke media



penyimpanan. Fungsi *callback* yang digunakan pada skenario kedua dapat dilihat pada gambar 5.12

| Pseudocode fungsi untuk menangani data |   |
|--|---|
|  | <pre> DEFINE on_response()   DO send received data   FOR each topic in topic_list     DO flush topic   END </pre> |

**Gambar 5.12 Pseudocode fungsi *callback* untuk menangani data yang diterima dari *middleware***

### 5.2.3 Implementasi pengiriman data dengan HTTP POST

Agar internet *gateway device* dapat mengirimkan data dari *middleware* ke media penyimpanan yang berada di internet maka diperlukan program untuk mengirimkan data menggunakan protokol HTTP melalui koneksi GPRS. Sebelum dilakukan pengiriman, internet *gateway device* perlu dihubungkan terlebih dahulu dengan internet menggunakan koneksi GPRS, untuk itu pada sistem yang diajukan, digunakan program ppp untuk membuat koneksi internet, untuk mengaktifkan ppp digunakan perintah dalam program seperti pada gambar 5.13

|                         |
|-------------------------|
| call.sh                 |
| #!/bin/bash<br>pon rnet |

**Gambar 5.13 Program untuk melakukan koneksi internet**

Pada penelitian ini digunakan penyedia layanan Telkomsel sebagai operator untuk jaringan seluler. Konfigurasi yang digunakan untuk melakukan koneksi jaringan seluler dapat dilihat pada gambar 5.14.

|   |
|---|
| rnet  |
| "/usr/sbin/chat -v -f /etc/chatscripts/gprs -T<br>internet"<br>/dev/ttyAMA0<br>9600<br>noipdefault<br>usepeerdns<br>defaultroute<br>persist<br>noauth<br>nocrtscts<br>local |

**Gambar 5.14 Konfigurasi program ppp**

APN atau *access point name* yang digunakan adalah internet, jika digunakan penyedia layanan lain APN dapat disesuaikan dengan konfigurasi setiap penyedia layanan. Selain itu digunakan juga program untuk memutus koneksi yang disediakan program ppp, perintah untuk memutus koneksi, terdapat pada program seperti pada gambar 5.15

|                          |
|--------------------------|
| uncall.sh                |
| #!/bin/bash<br>poff rnet |



### Gambar 5.15 Program untuk memutus koneksi internet

Implementasi pengiriman data dengan HTTP POST juga dibagi dua berdasarkan skenario yang diajukan.

#### 5.2.3.1 Skenario dengan rentang waktu subscribe

| Pseudocode program untuk melakukan koneksi |  |
|--|--|
|  | DEFINE connect()<br>DO subprocess.open(call.sh)<br>DO sleep 20 seconds |

Gambar 5.16 Pseudocode untuk melakukan koneksi internet

Untuk skenario dengan rentang waktu *subscribe* dibutuhkan proses pengiriman data yang telah disimpan pada *file* JSON. Sebelum pengiriman data dilakukan perlu dilakukan koneksi ke internet dengan program untuk melakukan koneksi internet seperti pada gambar 5.13, untuk memanggil program tersebut pada program untuk mengirim data dengan HTTP digunakan fungsi seperti pada gambar 5.16

| Pseudocode pengiriman data |  |
|----------------------------|--|
|                            | DEFINE sendHTTP()<br>DO open JSON <i>file</i> as <i>file</i><br>SET data_json=json.load( <i>file</i> )<br>SET data = json.dumps(data_json)<br>DO make GPRS connection<br>DO connect(http://iot.dijalinbersari.com:5001)<br>SET headers = {"Content-type" : "application/json"}<br>DO post(http://iot.dijalinbersari.com:5001/api/post,data,headers)<br>DO getResponse()<br>DO delete <i>file</i> |

Gambar 5.17 Pseudocode mengirim data dengan penyimpanan sementara

Data yang diterima dari *middleware* dikirim menggunakan *method* HTTP POST yang dikirim ke alamat `http://iot.dijalinbersari.com:5001/api/post`. Sebelumnya data yang akan dikirim diambil dari *file* JSON yang telah disimpan ketika proses *subscribe*. Pada skenario pertama, setiap dilakukan proses pengiriman *file* ke media penyimpanan, dilakukan proses koneksi ke internet menggunakan GPRS. Setelah proses pengiriman selesai dan respon dari media penyimpanan diterima oleh *internet gateway device*, *file* JSON yang sebelumnya dipakai untuk menyimpan data sementara akan dihapus. Untuk mengimplementasikan seluruh tahapan pengiriman data digunakan program seperti pada gambar 5.17.

#### 5.2.3.2 Skenario pengiriman langsung

Skenario pengiriman langsung menggunakan koneksi GPRS yang terus menyala sehingga setiap pengiriman data menggunakan HTTP POST tidak diperlukan lagi pembentukan koneksi GPRS.

| Pseudocode pengiriman langsung |                       |
|--------------------------------|-----------------------|
|                                | DEFINE sendHTTP(data) |

|  |
|--|
| <pre>SET data = json.dumps(data) DO connect(http://iot.dijalinbersari.com:5001) SET headers = {"Content-type" : "application/json"} DO post(http://iot.dijalinbersari.com:5001/api/post,data,headers) DO getResponse()</pre> |
|--|

**Gambar 5.18 Pseudocode untuk mengirim data dengan pengiriman langsung**

Karena setiap data yang diterima dari *middleware* kemudian akan langsung dikirimkan menggunakan *method* HTTP POST ke alamat `http://iot.dijalinbersari.com:5001/api/post` data yang dikirim diperoleh dari *callback* sehingga tidak diperlukan adanya mekanisme pengambilan data dari *file* JSON, *pseudocode* yang digunakan untuk pengiriman data langsung dapat dijelaskan pada gambar 5.18



## BAB 6 PENGUJIAN DAN PEMBAHASAN HASIL PENGUJIAN

Bab pengujian dan pembahasan hasil pengujian berisikan tentang langkah-langkah pengujian yang dilakukan untuk memastikan kebutuhan yang dideskripsikan sebelumnya telah tercapai. Pengujian penelitian ini dibagi menjadi dua yaitu pengujian fungsional yang menguji fungsionalitas sistem internet *gateway device* berdasarkan kebutuhan fungsional, dan pengujian nonfungsional untuk menguji kinerja internet *gateway device* sesuai dengan kebutuhan nonfungsional

### 6.1 Pengujian Fungsional

Pengujian fungsional dilakukan dengan menjalankan skenario pengujian yang telah dideskripsikan sebelumnya pada perancangan pengujian. Pengujian fungsional dikatakan valid apabila hasil yang diharapkan sesuai dengan hasil pengujian yang dilakukan.

#### 6.1.1 Pengujian *subscribe* tanpa definisi topik

|                       |  |
|-----------------------|--|
| Nama Kasus Uji        | <i>Subscribe</i> semua <i>topic</i>  |
| Tujuan Pengujian      | Menguji Internet <i>Gateway Device</i> untuk <i>subscribe</i> semua <i>topic</i> dari <i>middleware</i>  |
| Prosedur Pengujian    | 1. <i>Middleware</i> sudah berjalan<br>2. Pengguna menjalankan program berbasis python untuk melakukan <i>subscribe</i> pada <i>middleware</i> dengan <i>topic</i> yang didapatkan dari <i>redis key</i> |
| Hasil yang diharapkan | Internet <i>gateway device</i> dapat <i>mensubscribe</i> semua <i>topic</i> dari <i>middleware</i>   |
| Hasil pengujian       | Internet <i>gateway device</i> berhasil <i>mensubscribe</i> semua <i>topic</i> dari <i>middleware</i>  |

```
pi@TheMiddleware:~/sendHTTP/fix $ python sendGSMwait.py
subscribe ke topic office/roomA13
subscribe ke topic office/roomA14
subscribe ke topic office/roomA15
subscribe ke topic office/gambarA13
menerima data...
jumlah data : 1
```

Gambar 6.1 Hasil pengujian *subscribe* ke semua topik

Hasil pengujian *subscribe* ke semua topik dapat dilihat di gambar 6.1. Dari pengujian *subscribe* semua *topic*, program dapat melakukan *subscribe* ke topik yang diambil dari *middleware*. Hasil uji yang didapatkan menunjukkan bahwa internet *gateway device* melakukan *subscribe* ke empat buah topik sekaligus yaitu *office/roomA13*, *office/roomA14*, *office/roomA15* dan *office/gambarA13*.

### 6.1.2 Pengujian *subscribe* topik yang telah didefinisikan

|                       |  |
|-----------------------|--|
| Nama Kasus Uji        | <i>Subscribe satu topic</i>  |
| Tujuan Pengujian      | Menguji Internet Gateway Device untuk <i>subscribe</i> satu <i>topic</i> dari <i>middleware</i>  |
| Prosedur Pengujian    | <ol style="list-style-type: none"> <li>1. <i>Middleware</i> sudah berjalan</li> <li>2. Pengguna menjalankan program berbasis python untuk melakukan <i>subscribe</i> pada <i>middleware</i> dengan <i>topic</i> yang ditentukan dari parameter saat menjalankan program</li> </ol> |
| Hasil yang diharapkan | Internet gateway device dapat <i>mensubscribe</i> satu <i>topic</i> dari <i>middleware</i>   |
| Hasil pengujian       | Internet gateway device berhasil <i>mensubscribe</i> satu <i>topic</i> dari <i>middleware</i>  |

```

pi@TheMiddleware:~/sendHTTP/fix $ python sendGSMwait.py office/roomA14
subscribe ke topic office/roomA14
menerima data...
jumlah data : 1

```

**Gambar 6.2 Hasil pengujian *subscribe* topik yang telah didefinisikan**

Gambar 6.2 menunjukkan hasil pengujian *subscribe* ke satu topik yang dilakukan berdasarkan skenario yang ada. Hasil pengujian *subscribe* ke satu topik menunjukkan bahwa internet *gateway device* dapat melakukan *subscribe* ke topik yang didapatkan dari input yang diberikan ketika menjalankan program. Pengujian *subscribe* dengan topik yang telah didefinisikan, menggunakan topik *office/roomA14* yang telah didefinisikan terlebih dahulu pada program. Dari hasil uji yang didapatkan, internet *gateway device* melakukan *subscribe* ke satu topik yaitu *office/roomA14*.

### 6.1.3 Pengujian penerimaan data JSON

|                       |   |
|-----------------------|---|
| Nama Kasus Uji        | Menerima data JSON  |
| Tujuan Pengujian      | Menguji Internet <i>Gateway Device</i> untuk menerima data dari <i>middleware</i> dalam format JSON   |
| Prosedur Pengujian    | <ol style="list-style-type: none"> <li>1. <i>Middleware</i> sudah berjalan</li> <li>2. Pengguna menjalankan program berbasis python untuk menerima data JSON dengan melakukan <i>subscribe</i> pada <i>middleware</i> dengan <i>topic</i> yang diperoleh dari <i>redis key</i></li> </ol> |
| Hasil yang diharapkan | Internet <i>Gateway Device</i> dapat menerima data dari <i>middleware</i> dalam format JSON   |
| Hasil pengujian       | Internet <i>gateway device</i> berhasil menerima data dari <i>middleware</i> dalam format JSON  |

```

pi@TheMiddleware:~/sendHTTP/fix $ python sendGSMwait.py office/roomA14
subscribe ke topic office/roomA14
menerima data...
jumlah data : 1
payload {'protocol': 'coap', 'temperature': {'value': '24.6.6', 'unit': 'celcius'}, 'times
tamp': 1524107632, 'humidity': {'value': '53.4.4', 'unit': '%'}, 'topic': 'office/roomA14'
, 'sensor': {'index': 16446677, 'tipe': 'esp8266', 'module': 'dht22', 'ip': '192.168.42.12
'}}

```

**Gambar 6.3 Hasil pengujian penerimaan data JSON**

Setelah dilakukan pengujian, diperoleh hasil seperti ditunjukkan pada gambar 6.3. Dari hasil pengujian penerimaan data JSON, internet *gateway device* dapat menerima data dari topik *office/roomA14* dalam format JSON. Data yang didapatkan merupakan *payload* dari sensor DHT22 dalam bentuk teks.

### 6.1.4 Pengujian pengiriman data secara berkala

|                    |   |
|--------------------|---|
| Nama Kasus Uji     | Mengirimkan data secara berkala   |
| Tujuan Pengujian   | Menguji Internet <i>Gateway Device</i> untuk menyimpan sementara <i>file</i> yang diterima dari <i>middleware</i> dalam <i>file</i> JSON lalu mengirimkan ke media penyimpanan secara berkala |
| Prosedur Pengujian | <ol style="list-style-type: none"> <li>1. <i>Middleware</i> sudah berjalan</li> </ol>   |

|                       |   |
|-----------------------|---|
|                       | 2. Pengguna menjalankan program berbasis python untuk menerima data <i>File</i> dengan melakukan <i>subscribe</i> pada <i>middleware</i> dengan <i>topic</i> yang diperoleh dari <i>redis key</i> |
| Hasil yang diharapkan | Internet <i>Gateway Device</i> dapat menyimpan sementara <i>file</i> yang diterima dari <i>middleware</i> dalam <i>file</i> JSON lalu mengirimkan ke media penyimpanan secara berkala             |
| Hasil pengujian       | Internet <i>gateway device</i> berhasil menyimpan sementara <i>file</i> yang diterima dari <i>middleware</i> dalam <i>file</i> JSON lalu mengirimkan ke media penyimpanan secara berkala          |

```

menerima data...
jumlah data : 13
13 data
2381767 bytes
sending at : 1524990980.46
DATA TEST TERKIRIM

```

**Gambar 6.4 Hasil pengujian pengiriman data secara berkala**

Pengujian pengiriman data secara berkala menunjukkan hasil seperti pada gambar 6.4. Hasil pengujian pengiriman data secara berkala menunjukkan internet *gateway device* dapat mengirimkan data yang diterima dari *middleware* ke media penyimpanan berbasis *cloud*. Hal ini dapat dilihat dari respon yang diterima dari media penyimpanan berupa pesan DATA TEST TERKIRIM. Pada saat pengujian, data yang diterima pada internet *gateway device* selama dilakukan *subscribe* sebanyak 13 data dengan besar data 2381767 bytes. Hasil pengujian juga menampilkan waktu internet *gateway device* mulai mengirimkan data ke media penyimpanan yaitu pada 1524990980 yang merupakan waktu dalam format epoch.

#### 6.1.5 Pengujian pengiriman data secara langsung

|                    |  |
|--------------------|--|
| Nama Kasus Uji     | Mengirimkan data secara langsung   |
| Tujuan Pengujian   | Menguji Internet <i>Gateway Device</i> untuk mengirimkan data yang diterima dari <i>middleware</i> ke media penyimpanan secara real time |
| Prosedur Pengujian | 1. <i>Middleware</i> sudah berjalan  |



|                       |   |
|-----------------------|---|
|                       | 2. Pengguna menjalankan program berbasis python untuk menerima data <i>File</i> dengan melakukan <i>subscribe</i> pada <i>middleware</i> dengan <i>topic</i> yang diperoleh dari <i>redis key</i> |
| Hasil yang diharapkan | Internet <i>Gateway Device</i> dapat mengirimkan data yang diterima dari <i>middleware</i> ke media penyimpanan secara real time  |
| Hasil pengujian       | Internet <i>gateway device</i> berhasil mengirimkan data yang diterima dari <i>middleware</i> ke media penyimpanan secara real time   |

```

pi@TheMiddleware:~/sendHTTP/fix $ python sendGSM.py office/roomA14
subscribe ke topic office/roomA14
menerima data...
jumlah data : 1
253 bytes
sending at : 1524110530.56
DATA TEST TERIKIRIM

```

**Gambar 6.5 Hasil pengujian pengiriman langsung**

Pengujian pengiriman langsung memberikan hasil seperti pada gambar 6.5. Dari hasil pengujian pengiriman langsung, diketahui bahwa internet *gateway device* dapat mengirimkan satu data yang diterima dari topik *office/roomA14* dengan besar data 253 *bytes* yang diketahui dari respon yang diterima dari media penyimpanan berupa pesan DATA TEST TERIKIRIM. Selain itu, juga ditampilkan bahwa data dikirimkan pada waktu 1524110530 yang merupakan waktu dalam format epoch.

#### 6.1.6 Pengujian pengiriman data dengan HTTP header

|                    |  |
|--------------------|--|
| Nama Kasus Uji     | Mengirimkan data menggunakan HTTP header   |
| Tujuan Pengujian   | Menguji Internet <i>Gateway Device</i> untuk mengirimkan data yang diterima dari <i>middleware</i> menggunakan HTTP header                           |
| Prosedur Pengujian | <ol style="list-style-type: none"> <li>1. <i>Middleware</i> sudah berjalan</li> <li>2. Pengguna menjalankan program berbasis python untuk</li> </ol> |

|                       |   |
|-----------------------|---|
|                       | menerima data <i>File</i> dengan melakukan <i>subscribe</i> pada <i>middleware</i> dengan <i>topic</i> yang diperoleh dari <i>redis key</i> |
| Hasil yang diharapkan | Internet <i>Gateway Device</i> dapat mengirimkan data yang diterima dari <i>middleware</i> menggunakan HTTP header                          |
| Hasil pengujian       | Internet <i>gateway device</i> berhasil mengirimkan data yang diterima dari <i>middleware</i> menggunakan HTTP header                       |

```

pi@TheMiddleware:~/sendHTTP/fix $ python sendGSM.py office/room14
menerima data... 233
258 bytes
sending at : 15241110549.01
HTTP HEADER : {'Content-type': 'application/json'}
DATA TEST TERIKIRIM

```

**Gambar 6.6 Hasil pengujian pengiriman dengan HTTP header**

Setelah dilakukan pengujian pengitiman degan HTTP header, didapatkan hasil pengujian seperti pada gambar 6.6. Hasil pengujian pengiriman dengan HTTP header menunjukkan bahwa internet *gateway device* dapat mengirimkan satu buah data berukuran 258 *bytes* yang diterima dari topik *office/room14* yang ditunjukkan dari respon berupa pesan DATA TEST TERKIRIM yang dikirimkan oleh media penyimpanan . Ditampilkan bahwa data dikirimkan dengan HTTP header 'Content-type' : 'application/json' pada waktu 1524110549 yang merupakan format waktu dalam epoch.

## 6.2 Pengujian Nonfungsional

Pengujian nonfungsional dilakukan dengan mengamati data yang didapatkan dari hasil menjalankan program, program *monitoring* vnstat dan hasil tangkapan wireshark. Pengujian fungsional dilakukan untuk memastikan bahwa implementasi yang dilakukan sudah dapat memenuhi kebutuhan nonfungsional yang telah dideskripsikan sebelumnya.

### 6.2.1 Pengujian *delay* dan *throughput*

Pengujian dilakukan dengan dua skenario, skenario pertama yaitu dengan melakukan *subscribe* ke *middleware* selama rentang waktu tertentu dan menyimpan data yang diterima pada sebuah *file* terlebih dahulu sebelum dikirimkan ke media penyimpanan. Rentang waktu yang digunakan pada penelitian ini adalah 225 detik, 450 detik, 900 detik dan 1800 detik. Dengan hasil yang diperoleh dapat dilihat pada tabel 6.1. Untuk hasil lama waktu *delay*

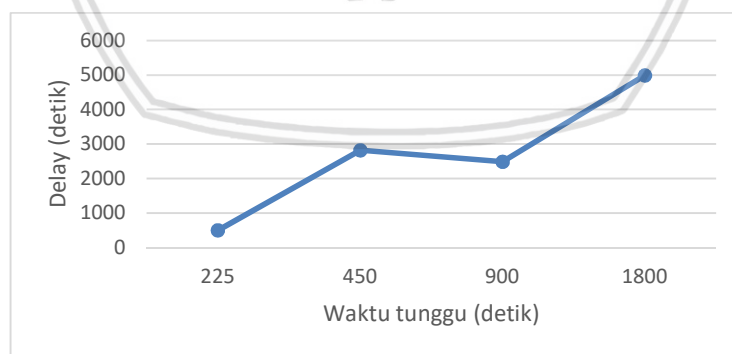
pengiriman data dapat dilihat pada gambar 6.7 sedangkan besar *throughput* yang diperoleh dapat dilihat pada gambar 6.8. Dari skenario pertama diperoleh rata-rata *throughput* sebesar 915,05 *bytes* per second.

Untuk skenario kedua, pengujian dilakukan dengan melakukan *subscribe* ke *middleware* kemudian data yang diperoleh akan dikirimkan secara langsung tanpa disimpan terlebih dahulu. Pada skenario ini digunakan 3 topik yang akan *unsubscribe*, yaitu *office/roomA13*, *office/roomA14*, dan *office/roomA15*. Dengan topik *office/roomA13* diterima data teks sebesar 253 *bytes* dan *delay* pengiriman selama 4 detik, untuk topik *office/roomA14* diterima data teks sebesar 258 *bytes* dan dengan *delay* pengiriman selama 4 detik, dan untuk topik *office/roomA15* diterima data gambar dengan ukuran 825,21 kilobytes dan dengan *delay* pengiriman selama 923 detik. Sedangkan hasil pengukuran *throughput* dari pengiriman 3 topik tersebut dapat dilihat pada gambar 6.9.

Data pengujian diperoleh dari program untuk melakukan *subscribe* serta pengiriman data yang menampilkan waktu ketika mengirim data dalam format epoch, ukuran data dalam satuan *bytes* serta jumlah data yang akan diteruskan. Untuk menentukan *delay* diperlukan data dari log web *service* / API dari media penyimpanan berbasis *cloud* berupa waktu data diterima dari internet *gateway* device.

**Tabel 6.1 Hasil pengujian *subscribe* dengan rentang waktu**

| rentang waktu | waktu kirim | Ukuran data ( <i>bytes</i> ) | Jumlah data | waktu diterima | <i>Delay</i> (detik) | <i>Throughput</i> (bps) |
|---------------|-------------|------------------------------|-------------|----------------|----------------------|-------------------------|
| 225           | 1524997987  | 453015                       | 6           | 1524998484     | 497                  | 911.49                  |
| 450           | 1525160250  | 2574984                      | 27          | 1525163068     | 2818                 | 913.76                  |
| 900           | 1525203034  | 2280521                      | 52          | 1525205520     | 2486                 | 917.34                  |
| 1800          | 1525210787  | 4581569                      | 105         | 1525215780     | 4993                 | 917.59                  |



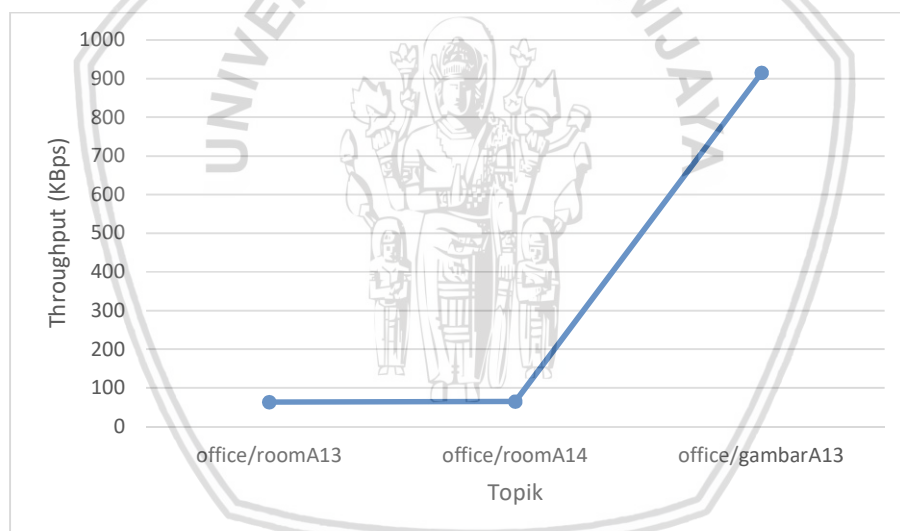
**Gambar 6.7 Grafik hasil pengujian *delay* dengan rentang waktu**

Dari hasil pengujian *delay* dan *throughput* pada skenario yang menggunakan rentang waktu, *delay* yang didapatkan tidak selalu berbanding lurus dengan rentang waktu yang digunakan untuk melakukan *subscribe*. Hal ini

dikarenakan besar data yang didapatkan tidak selalu berbanding lurus dengan lama waktu yang digunakan untuk melakukan *subscribe*. Bervariasinya besar data dikarenakan jenis data yang dikirimkan memiliki dua tipe yaitu data teks dan gambar yang berukuran relatif lebih besar dari data teks. Sehingga walaupun jumlah data yang didapatkan akan selalu bertambah seiring dengan bertambahnya waktu untuk melakukan *subscribe*, ukuran data yang didapatkan tidak selalu lebih besar dan mempengaruhi *delay* dalam pengiriman data ke media penyimpanan. Diluar itu, nilai *delay* yang didapatkan akan selalu bertambah seiring dengan bertambahnya ukuran data yang dikirimkan. Selain itu, hasil pengujian *throughput* didapatkan nilai *throughput* yang cenderung tetap yaitu antara 911,49 hingga 917,59 bytes per second.

**Tabel 6.2 Hasil pengujian pengiriman langsung**

| Topik            | ukuran | data   | delay | throughput |
|------------------|--------|--------|-------|------------|
| office/roomA13   | 698317 | 1 data | 739 s | 944.94     |
| office/roomA14   | 292    | 1 data | 2 s   | 146        |
| office/gambarA13 | 262    | 1 data | 2 s   | 131        |



**Gambar 6.8 Grafik pengujian *throughput* pengiriman langsung**

Data yang didapatkan dari topik *office/roomA13* dan *office/roomA14* merupakan *payload* data sensor DHT11 dan DHT22 dalam format teks, untuk topik *office/roomA15* merupakan *payload* data sensor kamera dalam format gambar. Dari hasil pengujian tersebut didapatkan hasil *throughput* untuk pengiriman data gambar lebih tinggi daripada *throughput* untuk pengiriman data teks, hal ini dikarenakan untuk pengiriman data teks, *delay* yang didapatkan terlalu singkat sehingga kinerja pengiriman melalui koneksi GPRS tidak dapat mencapai hasil yang maksimal. Untuk mendapatkan hasil yang maksimal dari hasil pengujian ini, data yang diteruskan minimal harus memuat satu data gambar. Dari hasil pengujian skenario kedua rata-rata *throughput* yang dihasilkan adalah 347,75 bytes per detik atau lebih rendah

### 6.2.2 Pengujian *reliability*

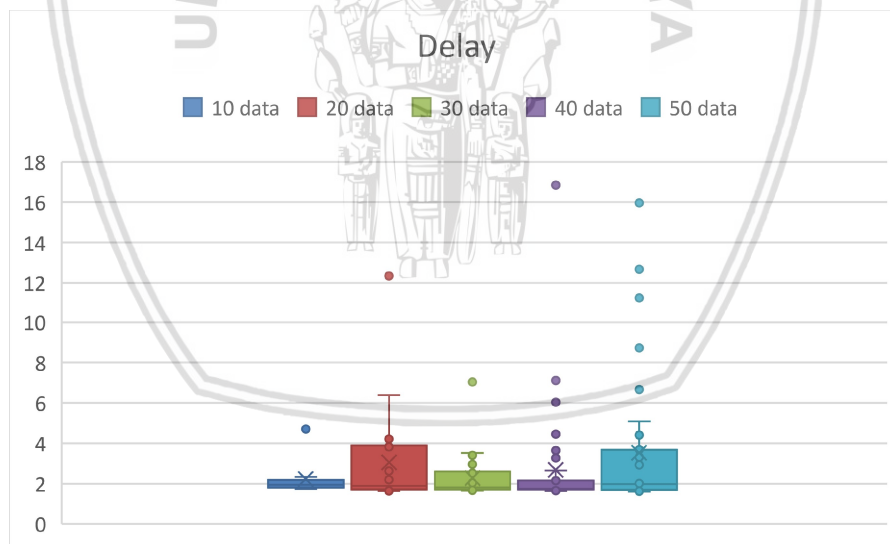
Pengujian *reliability* dilakukan dengan mengirimkan data dengan jumlah 10, 20, 30, 40, dan 50 data untuk tiga topik yang tersedia, yaitu *office/roomA13*, *office/roomA14* dan *office/gambarA13*. Hasil akurasi pengiriman data untuk masing-masing topik dapat dilihat pada tabel 6.3.

**Tabel 6.3 Hasil pengujian *reliability***

| topik                   | 10 data | 20 data | 30 data | 40 data | 50 data |
|-------------------------|---------|---------|---------|---------|---------|
| <i>office/roomA13</i>   | 100%    | 100%    | 100%    | 100%    | 100%    |
| <i>office/roomA14</i>   | 100%    | 100%    | 100%    | 100%    | 100%    |
| <i>office/gambarA13</i> | 100%    | 100%    | 93.33%  | 85%     | 80%     |

Dari hasil pengujian diperoleh akurasi 100% ketika mengirimkan data dari topik *office/roomA13* dan *office/roomA14*, namun ketika mulai mengirimkan data dari topik *office/gambarA13* dengan jumlah 30 data, 40 data dan 50 data tingkat akurasi menurun menjadi 93.33% untuk 30 data, 85% untuk 40 data dan 80% untuk 50 data, hal ini diakibatkan ukuran data yang lebih besar jika dibandingkan dengan data dari topik *office/roomA13* dan *office/roomA14*.

Nilai sebaran *delay* untuk pengujian pengiriman data dari topik *office/roomA13* dapat dilihat pada gambar 6.9

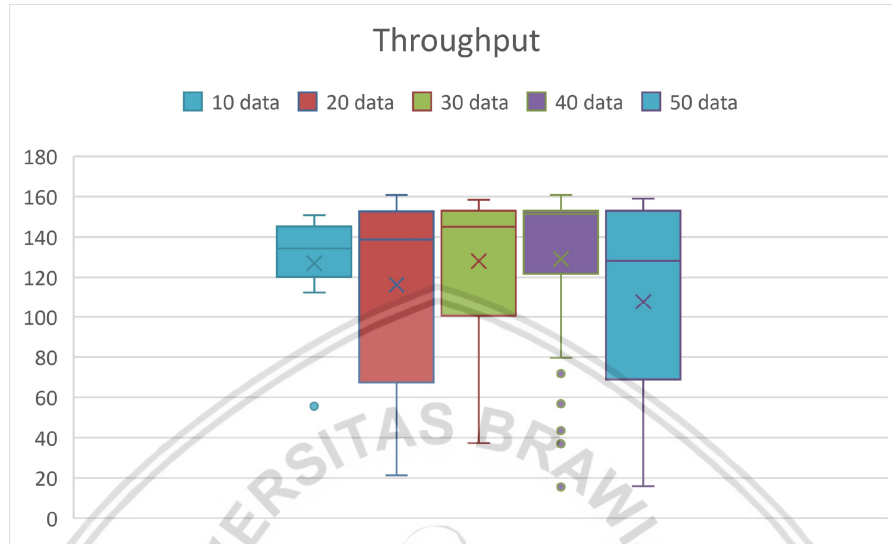


**Gambar 6.9 Grafik sebaran *delay* pengiriman data topik *office/roomA13***

Dari pengujian pengiriman dengan topik *office/roomA13* diperoleh rata-rata *delay* 2,23 detik untuk 10 data, 3,06 detik untuk 20 data, 2,28 detik untuk 30 data, 2,68 detik untuk 40 data dan 3,53 detik untuk 50 data, sehingga total rata-rata *delay* yang diperoleh adalah 2,75 detik.

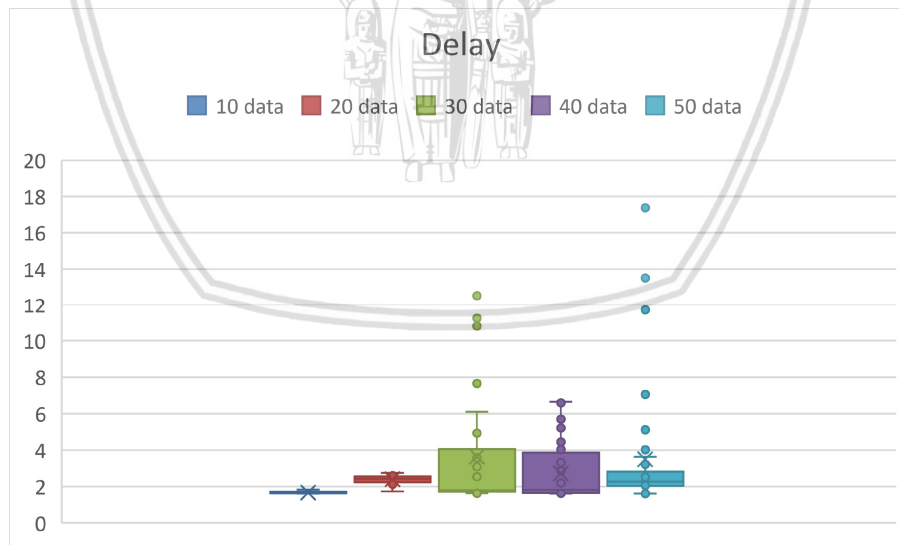
Untuk nilai sebaran *throughput* pengiriman dengan topik *office/roomA13* dapat dilihat pada gambar 6.10. Dari pengujian pengiriman dengan topik *office/roomA13* diperoleh *throughput* dengan rata-rata 126,95 bytes per detik

untuk pengiriman 10 data, 116,33 *bytes* per detik untuk 20 data, 128,1 *bytes* per detik untuk 30 data, 129,09 *bytes* per detik untuk 40 data, dan 107,76 *bytes* per detik untuk pengiriman 50 data, sehingga diperoleh rata-rata *throughput* dari seluruh pengiriman sebesar 121,65 *bytes* per detik.



**Gambar 6.10** Grafik sebaran *throughput* pengiriman data topik *office/roomA13*

Dari pengujian pengiriman data untuk topik *office/roomA14* diperoleh sebaran *delay* yang dapat dilihat pada gambar 6.11.

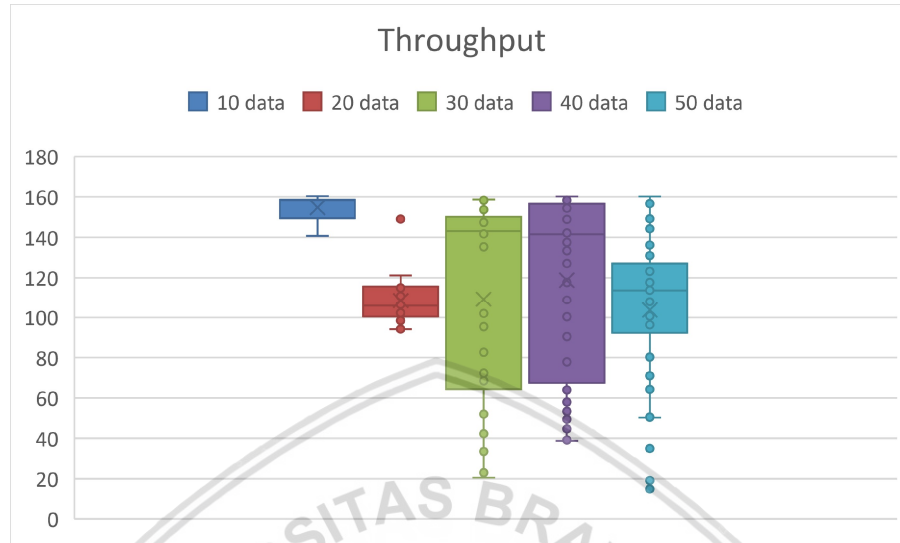


**Gambar 6.11** Grafik sebaran *delay* pengiriman data topik *office/roomA14*

Rata-rata *delay* untuk pengiriman data dari topik *office/roomA14* untuk 10 data adalah sebesar 1,86 detik, 2,4 detik untuk 20 data, 3,63 detik untuk 30 data, 2,71 detik untuk 40 data dan 3,49 detik untuk 50 data sehingga didapatkan rata-rata dari semua pengiriman sebesar 2,78 detik.

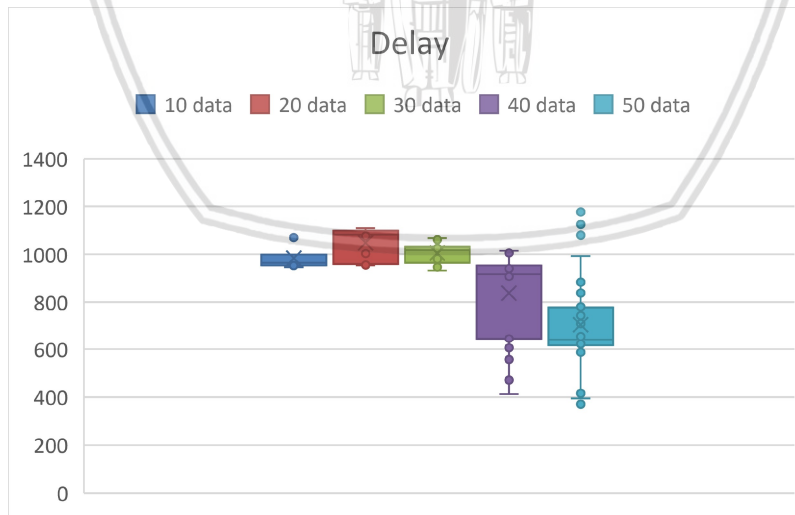


Sebaran *delay* untuk pengiriman data dari topik *office/roomA14* dapat dilihat pada gambar 6.12.



**Gambar 6.12** Grafik sebaran *throughput* pengiriman data topik *office/roomA14*

Dari pengujian pengiriman data untuk topik *office/roomA14* diperoleh rata-rata *throughput* untuk 10 data sebesar 154,83 bytes per detik, 110,1 bytes per detik untuk 20 data, 84,65 bytes per detik untuk 30 data, 135,19 bytes per detik untuk 40 data dan 105,85 bytes per detik untuk 50 data, sehingga dari semua pengiriman diperoleh rata-rata *throughput* sebesar 118,12 bytes per detik.



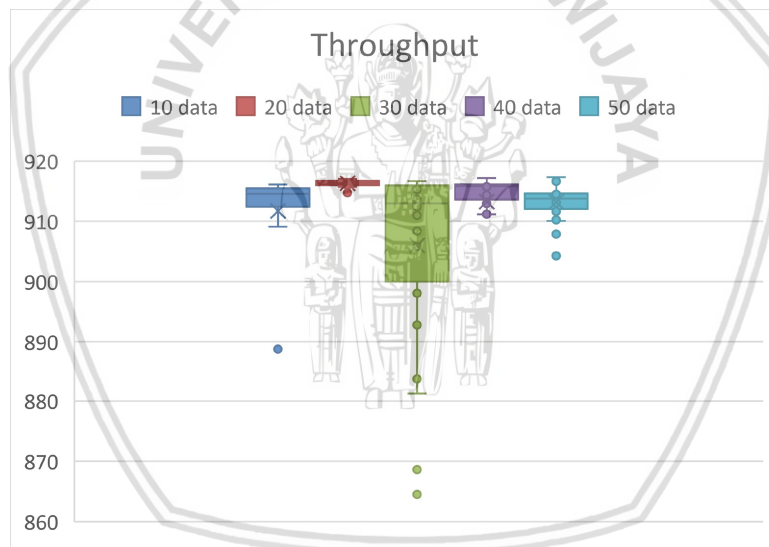
**Gambar 6.13** Grafik sebaran *delay* pengiriman data topik *office/gambarA13*

Sedangkan untuk pengujian pengiriman data dengan topik *office/gambarA13* diperoleh sebaran nilai *delay* yang dapat dilihat pada gambar 6.13. Dari hasil pengujian pengiriman data topik *office/roomA13* diperoleh

rata-rata *delay* untuk pengiriman 10 data sebesar 384,47 detik, 1048,2 detik untuk pengiriman 20 data, 1005,6 detik untuk 30 data, 837,98 detik untuk 40 data, dan 706,3 detik untuk pengiriman 50 data, sehingga diperoleh rata-rata *delay* dari semua pengiriman sebesar 916,52 detik.

Untuk sebaran nilai *throughput* pengiriman data topik *office/gambarA13* dapat dilihat pada gambar 6.14. Dari hasil pengujian diperoleh rata-rata *throughput* untuk 10 data sebesar 912,15 bytes per detik, 916,3 bytes per detik untuk 20 data, 906,01 bytes per detik untuk 30 data, 913,26 bytes per detik untuk 40 data dan 913,22 bytes per detik untuk 50 data, sehingga rata-rata *throughput* untuk semua pengiriman sebesar 912,19 bytes per detik.

Dari seluruh pengujian scalability dapat diketahui bahwa pengiriman data berupa gambar dari topik *office/gambarA13* dapat menghasilkan *throughput* yang lebih optimal dikarenakan ukuran data yang lebih besar dibandingkan data dari topik *office/roomA13* dan *office/roomA14*. Namun akurasi data yang didapatkan dari pengiriman data topik *office/gambarA13* lebih rendah ketika mengirimkan data sebanyak 30, 40 dan 50 data jika dibandingkan dengan pengiriman data topik *office/roomA13* dan *office/roomA14*.



Gambar 6.14 Grafik sebaran *throughput* pengiriman data topik *office/gambarA13*

### 6.2.3 Pengujian scalability

Tabel 6.4 Hasil pengujian scalability

| ukuran data | <i>delay</i> |
|-------------|--------------|
| 1MB         | 1144 detik   |
| 2MB         | 2287 detik   |
| 5MB         | 5696 detik   |
| 10MB        | 11398 detik  |

Pengujian *scalability* dilakukan dengan mengirimkan data dengan ukuran 1MB, 2MB, 5MB dan 10MB. Data diperoleh dari hasil *capture* wireshark yang dijalankan selama proses pengiriman data, dari data tersebut diperoleh hasil *delay* seperti pada tabel 6.4. Dari hasil pengujian yang diperoleh, sistem masih dapat menangani data sebesar 10MB namun dengan *delay* selama 11398 detik atau selama 3 jam 9 menit 58 detik.



## BAB 7 KESIMPULAN DAN SARAN

### 7.1 Kesimpulan

Dari penelitian yang berjenis implementatif pengembangan ini, penulis kemudian dapat mengambil kesimpulan berdasarkan rangkaian proses penelitian yang telah dilakukan. Kesimpulan yang dapat diambil antara lain :

1. Sistem internet *gateway device* dapat dibangun untuk mendukung fungsi penerimaan data dari *middleware* dan pengiriman data ke media penyimpanan berbasis *cloud* menggunakan perangkat raspberry pi dengan tambahan modul GSM sebagai penyedia layanan internet melalui koneksi GPRS.
2. Sistem internet *gateway device* dapat diterapkan pada lingkungan IoT untuk menerima data dari *middleware* dan mengirim data ke media penyimpanan, hal ini berdasarkan hasil pengujian fungsional yang didapatkan yaitu:
  - a. Internet *gateway device* dapat melakukan *subscribe* ke *middleware* tanpa didefinisikan topik terlebih dahulu.
  - b. Internet *gateway device* dapat melakukan *subscribe* ke *middleware* dengan topik yang telah didefinisikan.
  - c. Internet *gateway device* dapat menerima data dari *middleware* dalam format JSON.
  - d. Internet *gateway device* dapat menyimpan sementara file yang diterima dari *middleware* dalam file JSON lalu mengirimkan ke media penyimpanan secara berkala.
  - e. Internet *gateway device* dapat mengirimkan data yang diterima dari *middleware* ke media penyimpanan secara langsung.
  - f. Internet *gateway device* dapat mengirimkan data yang diterima dari *middleware* menggunakan HTTP *header*
3. Sistem internet *gateway device* memiliki kinerja yang baik, hal ini didapatkan dari hasil pengujian nonfungsional antara lain :
  - a. Rata-rata *throughput* yang didapatkan pada skenario pengiriman dengan waktu tunggu adalah 915,05 *bytes* per detik, sedangkan untuk pengiriman langsung didapatkan 347,75 *bytes* per detik.
  - b. Tingkat akurasi data dari pengujian *reliability* didapatkan nilai 80%-100% akurasi data dengan *packet loss* terjadi hanya pada pengiriman 30, 40, dan 50 data dengan topik office/gambarA13.
  - c. Pengiriman data internet *gateway device* memiliki skalabilitas yang baik berdasarkan hasil pengujian *scalability* yang dapat mengirimkan data hingga 10MB.

### 7.2 Saran

Dalam penelitian ini, penulis memberikan beberapa saran untuk mengembangkan ke penelitian selanjutnya, antara lain:

1. Digunakannya modul dengan teknologi jaringan seluler yang lebih bagus daripada modul GSM, misalnya modul LTE.
2. Dilakukannya analisis kinerja koneksi jaringan seluler yang lebih mendalam untuk mengetahui performa jaringan seluler ketika digunakan dalam internet *gateway device*.
3. Penambahan tipe data dari sensor yang lebih beragam untuk mengetahui pengaruhnya dalam kinerja internet *gateway device*.



## DAFTAR PUSTAKA

- Arganata, G., Pramukantoro, E.S. and Yahya, W., 2018. Pengembangan Sistem Penyimpanan Data Berbasis MongoDB dan GridFS Untuk Menyimpan Data Yang Beragam Dari *Node Sensor*. 2(7), pp.2549–2557.
- ECMA, I., 2017. The JSON Data Interchange Format. *ECMA International*, [online] 2st Editio(December), p.16. Available at: <<http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>>.
- Gouveia, F.C. De and Magedanz, T., 2011. Quality of Service in Telecommunication. *Telecommunication Systems and Technologies*, II, pp.1–8.
- Gubbi, J., Buyya, R., Marusic, S. and Palaniswami, M., 2013. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, [online] 29(7), pp.1645–1660. Tersedia di: <<http://dx.doi.org/10.1016/j.future.2013.01.010>>.
- Kurose, James (University of Massachusetts, A. and Ross, K. (Polytechnic I. of N., 2013. *Computer Networking - A top-down approach*.
- Ma, M., Wang, P. and Chu, C.-H., 2013. Data Management for Internet of Things: Challenges, Approaches and Opportunities. *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*, [online] pp.1144–1151. Tersedia di: <<http://ieeexplore.ieee.org/document/6682212/>>.
- Tan, L., 2010. Future internet: The Internet of Things. *2010 3rd International Conference on Advanced Computer Theory and Engineering(ICAETE)*, [online] pp.V5-376-V5-380. Tersedia di: <<http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5579543>>.
- Vejlgaard, B., Lauridsen, M., Nguyen, H., Kovacs, I.Z., Mogensen, P. and Sorensen, M., 2017. Coverage and Capacity Analysis of Sigfox, LoRa, GPRS, and NB-IoT. *IEEE Vehicular Technology Conference*, 2017–June.
- Zhu, Q., Wang, R., Chen, Q., Liu, Y. and Qin, W., 2010. IOT gateway: Bridging wireless sensor networks into Internet of Things. *Proceedings - IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, EUC 2010*, pp.347–352.